UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Implantación de un Esquema de Seguridad Informática

Cilene Arroyo Criado
Tutor: Álvaro Ortigosa Juárez
Enero 2018

Implantación de un Esquema de Seguridad Informática

AUTOR: Cilene Arroyo Criado
TUTOR: Álvaro Ortigosa Juárez

Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Enero de 2018

Resumen (castellano)

Este Trabajo Fin de Grado expondrá un ejemplo de modelo de cómo montar la seguridad informática de una empresa, además de explicar en profundidad los ataques más comunes que existen actualmente. Para llevarlo a cabo, se debe estudiar el funcionamiento de la empresa para decidir qué tipo de tecnologías se amoldan mejor a sus necesidades, de modo que se explicarán las diversas plataformas del mercado Firewalls, SIEMs, IDS y WAFs señalando cuales se ajustan a cada necesidad.

Previamente a montar un cortafuegos se debe concienciar de los riesgos que se correrán y cómo minimizar su impacto. Durante el proceso de implantación se encontrarán diversos problemas como, por ejemplo, tráfico legítimo bloqueado por una mala codificación por parte de desarrollo.

Una vez que todo esté implantado, se deberá configurar las reglas de correlación. Mediante estas reglas se controlarán, por ejemplo, los ataques SQLi, PHPi, XSS, File Intrusion, Brute Force, escaneo horizontal/vertical, uso del usuario administrador dentro de la empresa...etc. En el proyecto se expondrán todas las reglas básicas junto con otras más avanzadas. Finalmente, el proyecto se centrará en como configurar un firewall de aplicaciones web.

Abstract (English)

This Bachelor Thesis will present an example of how to assemble the computer security of a company, in addition it would be explained the most common attacks that currently exist. To carry it out, you must study the operation of the company to decide what kind of technologies are needed to meet the needs of the company. It would be explained the different platforms that exist as Firewalls, SIEM, IDS, WAF indicating that they do.

Before setting up a firewall you should be aware of the risks that will be run and how they will minimize their impact. During the implementation process several problems were found, for example, a legitimate fact blocked by bad coding by development.

Once everything is in place, you must configure the correlation rules. With these rules, de most important attacks will be blocked, for example, SQL attacks, PHP, XSS, File Intrusion, Brute Force, horizontal / vertical scanning, use of the administrator user within the company ... etc. In the project all the basic rules will be exposed along with other more advanced ones. Finally, the project will focus on how to configure a web application firewall.

Palabras clave (castellano)

Seguridad, correlación, intrusión, eventos, bloqueo, WAF, firewall, IPS, IDS, logs, alertas.

Keywords (inglés)

Security, correlation, intrusion, events, blocking, WAF, firewall, IPS, IDS, logs, alerts.

Agradecimientos

A mi madre, a sisi y a abo.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introducción

1.1 Motivación

La seguridad informática, también conocida como ciberseguridad o seguridad de tecnologías de la información, es el área relacionada con la informática y la telemática que se enfoca en la protección de la infraestructura, especialmente, la información contenida en un ordenador a través de las redes. Para ello existen una serie de estándares, protocolos, métodos, reglas, herramientas y leyes concebidas para minimizar los posibles riesgos a la infraestructura o a la información.

Garantiza en la medida de lo posible, ya que no existe la seguridad absoluta, las condiciones de los datos y la información de las empresas y particulares, manteniendo la confidencialidad, integridad y la disponibilidad de los servicios. La seguridad informática da protección contra las amenazas, evitando daños y minimizar los riesgos.

La seguridad está concebida para proteger los activos informáticos, abarcando principalmente las siguientes áreas:

- Infraestructura: en una empresa hay múltiples servidores y equipos, cada uno con una función, como almacenamiento de datos, servidor origen etc. Se deben tener en cuenta todas las arquitecturas de red para anticiparse a fallos, robos, vulnerabilidades etc.
- Clientes: todos aquellos clientes de la empresa podrán ser vulnerables ante un ataque, debido a que sus datos son almacenados en bases de datos, por lo que se debe tener una seguridad robusta.

Las amenazas de seguridad pueden proceder desde múltiples orígenes, desde programas dañinos instalados en el pc del usuario (virus) hasta programas que acceden de manera remota. En la actualidad existe una amplia variedad de virus que pueden vulnerar cualquier equipo entre otros:

- Virus residentes en memoria: Se alojan en la memoria del ordenador y se activan cuando el sistema operativo se ejecuta, infectando todo archivo que se abra. Teniendo así un control de la memoria del sistema.
- Virus directos: Cuando se cumple una acción específica el virus se activará, replicándose e infectando los ficheros.
- Virus de sobre escritura: Borran la información contenida de los ficheros que infecta.
- Virus de sector de arranque: Afecta al sector de arranque del disco duro.
- Macro Virus: Virus que suelen llegar por correo afectando a programas que contienen macros como .doc, .xls etc.
- Virus polimórfico: Se encriptan de manera distinta, lo que hacer casi imposible que los antivirus lo detectan. Utilizan algoritmos y claves de cifrados cada vez que infectan un nuevo sistema.
- Virus faat: Impiden el acceso a secciones del disco donde se almacena la información sobre la ubicación de los archivos, el espacio disponible, espacios de disco que no es deben utilizar etc.

- Virus de comandos: Páginas web que incluyen código que ejecutar ciertos comandos llevando a acciones maliciosas.

Las amenazas de seguridad pueden ser provocados desde diferentes fuentes:

- Errores de programación: Una mala programación provoca agujeros en los sistemas, pudiendo ser explotados por crackers. Suele ser una de las razones más comunes para encontrar vulnerabilidades.
- Programas maliciosos: Programas desarrollados para uso ilícito de los sistemas. Se instalan en el ordenador mediante inyecciones de código/comandos o virus informáticos, permitiendo ejecutar un archivo desde dentro. Esto permite conseguir/modificar datos.
- Usuarios: Un usuario final es el mayor problema, muchas veces por permisos demasiado elevados, accesos no restringidos etc.
- Hackers: Consiguen acceder a los datos y programas de manera no autorizada.
- Catástrofe natural: Terremotos, inundaciones, incendios etc.
- Personal interno: Técnicos que administran el sistema pueden hacerlo vulnerable mediante disputas internas, problemas laborales, despidos, fines lucrativos etc.

Los ataques pueden ser tanto internos, externos o un conjunto de ambos. La parte del sistema que está en el exterior abierto al público, abierto a Internet, da la posibilidad de que un atacante pueda entrar, pudiendo alterar el funcionamiento de la red. Sin embargo, que un sistema no esté conectado a Internet, no garantiza su seguridad.

Ataques internos:

- Personal interno conoce la red y el fin de cada equipo, por lo que saben la ubicación de los datos, datos sensibles para empresa.
- IPS y/o firewalls que están más orientados al análisis de tráfico que proviene del exterior. Sin embargo, no son configurados para analizar el tráfico interno con la misma exigencia, un grave error debido a que se podría acceder al interior desde redes inalámbricas desprotegidas, equipos sin vigilancia, routers sin capa de seguridad etc.

Ataques externos:

- Ataques fuera de la red, y al no tener la información certera de la infraestructura de la red, el atacante debe realizar ciertos intentos para poder qué hay detrás y buscar sus vulnerabilidades. Este tipo de ataques pueden ser prevenidos mediante múltiples herramientas.

Ataque interno y externo:

- Según el tipo de amenaza pueden provocar distintos tipos ataque como robo de información, destrucción de bases de datos, dejar sin servicio a los sistemas, suplantación de identidades etc.

En la actualidad existen diversas herramientas para intentar mantener la seguridad informática de los equipos, programas antivirus en los pcs, contraseñas de seguridad, firewalls, ids (sistemas de detección de intrusos).

La evolución del entorno, donde el intercambio de datos a través de las redes es algo habitual, se ha convertido en un medio para intentar conseguir recursos de manera ilegítima. Con el tiempo han ido apareciendo distintos vectores de ataque, pudiendo destacar:

- Phishing: Intento de suplantación de identidad
- Malware: Virus, Gusanos
- Ataques web: SQLi, XSS, PHPi…
- Intentos de intrusión
- etc.

En este proyecto se expondrán los principales ataques actuales a los que se enfrentan las empresas diariamente y las capacidades de actuación antes los ataques de las herramientas disponibles.

1.2 Objetivos

El objetivo de este proyecto es mostrar las amenazas a las que están expuestas las empresas y particulares. Se quiere medir la efectividad de la seguridad perimetral que se están implementando en los últimos años en las diferentes empresas. Se creará un entorno de prueba.

Además, se tienen como objetivo, exponer los tipos de ataques que existen actualmente. Explicando brevemente los tipos de ataque más comunes, cómo se realiza cada uno de ellos y la manera de explotar las vulnerabilidades, todo ello visto desde el punto de vista de un atacante. Por otro lado, se muestra la manera de protegerse ante posibles ataques y cómo evitarlos.

Se creará un entorno de prueba virtualizado. En este se aplicarán distintos vectores de ataques mediante tests de intrusión y se mediará el grado de la eficacia.

Se desplegarán varios desarrollos de software libre en un entorno virtualizado con el fin de medir la efectividad de lo que está en el mercado. Las tecnologías que se implementarán:

- Un firewall de Aplicación Web
- Un Correlador que almacene todos los eventos de seguridad

1.3 Organización de la memoria

La memoria se divide principalmente en cinco capítulos.

- Capítulo 2 - Estado del Arte: Se expone la historia de la seguridad informática.

- Capítulo 3 - Herramientas de Seguridad: Se exponen los principales tipos de herramientas que existen para proporcionar seguridad a una infraestructura. Firewalls, IDS/IPS, WAF, SIEM …
- Capítulo 4- Tipos de Ataque: Se explican los tipos de ataque más habituales.
- Capítulo 5 - Opciones de protección para Empresas: Se muestran las diferentes tecnologías disponibles para cada tipo de vulnerabilidad que se quiera proteger.
- Capítulo 6 - Intentos de Ataque: Se muestra cómo realizar cada tipo de ataque.
- Capítulo 7 - Defensa ante Vulnerabilidades del Entorno: Entorno de prueba virtualizado como ejemplo de una red de seguridad viable para montar en una empresa como defensa ante vulnerabilidades del entorno.

2 Estado del arte

2.1 Nacimiento de la Seguridad Informática

A partir de los años 80 el uso de ordenadores personales comienza a hacerse común, esto hizo que apareciera una preocupación por la integridad de los datos almacenados en ellos.

En la década de los 90 comienzan a aparecer los primeros virus, tomando así conciencia del peligro que suponía para los usuarios que lo usan para uso personal y equipos conectados a Internet. Se comienzan a realizar los primeros ataques hacia sistemas informáticos.

A partir del año 2000 las amenazas comenzaron a generalizarse y extenderse, tomándose por primera vez de manera seria la seguridad informática.

Con el uso masivo de Internet, la protección de la información se ha convertido en una necesidad, creándose así las técnicas de cifrado, firmas digitales etc.

Los primeros inicios de programas maliciosos fueron en 1959 en Bell por cuatro programadores, H. Douglas Mcllory, Robert Thomas Morris, Victor Vysottsky y Ken Thompson. Desarrollaron un juego que consistía en ocupar toda la memoria RAM de equipo del contrincante en el menor tiempo posible[9].

El primer virus que se lanzó fue contra una máquina de IBM Serie 600. Fue creado por Robert Thomas Morris en 1972 y se le denominó como Creeper. El programa consistía en emitir periódicamente por pantalla "I'm a creeper…catch me if you can!". Tras este ataque fue creado el primer antivirus denominado Reaper.

James P.Anderson fue uno de los pioneros en 1980 escribió un documento en el que se relata por primera vez la Seguridad Informática, asentando las bases de algo que a día de hoy parece algo normal, pero en aquella época parecía ciencia ficción. El documento se llamó, Computer Security Threat Monitoring and Surveillance[10] (Anexo A), describe la importancia del comportamiento enfocado hacia la seguridad informática.

En el documento ponen las bases de qué es una vulnerabilidad, derivando en extensiones donde se determina hechos como que una vulnerabilidad no es conocida ni explotada hasta que es descubierta.

En 1984 comenzaron a expandirse los virus, atacando al contenido de los disquetes. Además, empieza a hacerse común el uso del ordenador personal, comenzando así la preocupación de la protección de los datos almacenados. A partir de 1990 aparecen los virus que comienzan a atacar a PCs y equipos conectados a Internet, apareciendo en correos electrónicos, links…a finales de los 90 aparecen multitud de gusanos y malware lo que dio paso a comenzar a tener consciencia de que se deben empezar a tomar medidas al respecto a partir del 2000[11].

Para la protección de datos salió la necesidad de transformar los datos, es decir, encriptar, de modo que sea más seguro su almacenamiento. Para ello se cifra y descifran los datos, se realizan criptoanálisis y se usan firmas digitales y las CA (Autoridades de Certificaciones).

2.2 TOP ataques durante la historia

Principales ataques que ha habido durante la historia[12, 28, 29].

2.2.1 Yahoo – 1000 millones de usuarios

En 2016 Yahoo sufrió un ataque mediante el cual consiguieron más de 1000 millones de cuentas de usuario, números de teléfono, contraseñas y preguntas y respuestas de seguridad que no estaban cifradas.

2.2.2 Yahoo – 500 millones de usuarios

En 2014 Yahoo sufrió un ataque mediante el cual consiguieron más de 500 millones de cuentas de usuario, números de teléfono, contraseñas y preguntas y respuestas de seguridad que no estaban cifradas, igual que en 2016.

2.2.3 Friend Finder Network Inc – 400 millones de usuarios

Compañía que gestiona diferentes páginas de citas, fue atacada consiguiendo datos personales de usuarios, como correos electrónicos, patrones de navegación y compra y su orientación sexual.

2.2.4 El gran *hack* de EE.UU - 160 millones de usuarios

Este ciberataque no afecto únicamente a una compañía, afecto a una larga lista, entre ellas, ASDAQ, 7-Eleven, JC. Penney, JetBlue, Dow Jones y Global Payment. El ataque tuvo lugar durante 7 años desde 2005, en donde se robaron 160 millones de tarjetas bancarias de cliente. Cinco personas de origen ruso fueron acusadas y condenadas por el caso.

2.2.5 Adobe – 152 millones de usuarios

Adobe sufrió el robo de 152 millones de cuentas bancarias en 2013. Sin embargo, Adobe no reconoció nunca la cifra, ellos afirmaron que solo fueron robadas 38 millones de cuentas.

2.2.6 Ebay - 145 millones de usuarios

Ebay sufrió un ataque a sus bases de datos, en donde se almacenaban los usuarios de las páginas de comercio online. Esto obligó a más de 145 millones de usuarios a cambiar sus contraseñas.

2.2.7 Heartland - 130 millones de usuarios

El hacker Albert González fue acusado de coordinar el ataque que se llevó datos de 130 millones de tarjetas de débito y crédito de la multinacional de pagos Heartland Payment Systems. Sucedió en 2008, pero no se hizo público hasta mayo de 2009.

2.2.8 TJX - 94 millones de usuarios

En 2007 TJX hizo público que sufrieron un ataque informático que puso en peligro los datos bancarios de 94 millones de clientes entre sus cadenas de tiendas TJX: 94 millones de usuarios.

2.2.9 AOL - 92 millones de usuarios

Ataque que tuvo lugar desde dentro de la empresa en 2004. Un ingeniero de la compañía que fue despedido utilizó sus conocimientos de la empresa para infiltrarse dentro de la red interna. Robó la lista de correos de 92 millones de usuarios, vendiéndola posteriormente a un grupo de spammers.

2.2.10 Banco Central de Bangladés - 81 millones de dólares

Varios hackers lograron acceder a los sistemas del Banco Central de Bangladés. Intentaron transferir 81 millones de dólares a varios casinos de Filipinas. Sin embargo, un error ortográfico evitó la catástrofe, debido a que el nombre mal escrito de uno de los destinatarios levantó las alarmas y permitió bloquear el ataque planeado para obtener casi mil millones de dólares.

2.2.11 Sony PlayStation Network- 77 millones de usuarios

Sony sufrió un ataque que robó la información de 77 millones de cuentas de usuarios de servicios PlayStation en todo el mundo. Recompensó a los usuarios y recibió sanciones en varios países.

2.2.12 Veteranos de EEUU - 76 millones de usuarios

Robaron 76 millones de fichas personales de veteranos de guerra estadounidenses, mediante un disco duro que fue enviado al servicio técnico en 2009.

2.2.13 Target - 70 millones de usuarios

Ataque hacia la cadena de tiendas Target en el que robaron los números de tarjeta bancaria y claves de 40 millones de personas que utilizaron las tarjetas de crédito en alguna tienda Target a finales de 2013. Además, 30 millones de usuarios vieron vulnerados sus datos personales, teléfono móvil y dirección de email.

2.2.14 Bitfinex - 64 millones de bitcoins

El mayor operador mundial de intercambio de bitcoin basado en dólares, Bitfinex, radicado en Hong Kong, fue el objeto de este ataque. La cotización del bitcoin superior cayó un 23% en los días posteriores.

2.2.15 Evernote - 50 millones de usuarios

La compañía Evernote reacciono tan rápido que no hubo daños. En 2013 Evernote envió una notificación a sus usuarios para que cambiaran sus contraseñas antes indicios de que su red había sido hackeada.

2.2.16 DDoS a Play Station y Twitter, entre otros

Se trata del mayor ataque de DDoS producido hasta la fecha. Producido por la botnet Mirai, compuesta por cientos de miles de cámaras IP junto a otros dispositivos IoT, dejó fuera de juego a múltiples servicios de Internet, llegando a afectar a Play Station Network y Twitter, entre otros. Se sospecha que este ciberataque habría sido también una prueba de concepto para afectar al funcionamiento de los sistemas de voto electrónico de EE.UU., antes de las elecciones del 8 de noviembre.

2.2.17 Fallo en la implementación de la pila TCP en sistemas Linux

El fallo en la implementación de la pila TCP en sistemas Linux, permitió la infección de malware de forma remota y el secuestro de tráfico de usuarios con sistema operativo Android. Degradó las conexiones a la red mediante el protocolo HTTPS redirigiendo el tráfico.

2.2.18 Fallo en los procesadores Qualcomm

El fallo en procesadores Qualcomm permitió acceder a información cifrada sin que se activasen los mecanismos de borrados en millones de teléfonos. Esta vulnerabilidad en la generación de las claves de cifrado afectó aproximadamente al 60% de los móviles Android del mercado.

3 Herramientas de Seguridad

3.1 Cortafuegos (Firewall)

Los cortafuegos[25] son sistemas de defensa encargados de limitar y autorizar los flujos de tráfico que se originan entre distintas redes o subredes que componen las entidades públicas y privadas.

Son usados con el fin de:

- Segmentar las redes
- Autorizar o denegar tráfico
- Securizar la red
- Autorizar comunicación entre redes, llegando a un registro de todas las comunicaciones.
- Encaminamiento de paquetes dentro de la red

La creación de reglas en el firewall es la manera de que el firewall haga su labor. Estas reglas son configuradas por los administradores con el fin de autorizar o denegar comunicaciones.

Los modelos actuales de Firewall:

- NG Firewall: Aporta la funcionalidad básica de firewall y características muy concretas a nivel de IPS (detector de intrusos) y control de aplicaciones. Actúan como firewall perimetral aportando inteligencia para ataques en capa de aplicación.

- Firewall UTM: Firewalls que engloban diversas funciones en un único dispositivo. Diseñador para funcionar como un elemento de seguridad perimetral, IPS, antivirus, control de aplicaciones, proxy y QoS. Son modelos muy extendidos para PyMEs, debido a que una gran cantidad de dispositivos sería cara de obtener.

3.2 IPS/IDS

Dispositivos de tipo de detector de intrusos[26]. Acciones que intentan comprometer la integridad, disponibilidad y confidencialidad de un servicio.

Existen dos tipos de dispositivos, dependiendo de la ubicación del dispositivo en la red y el comportamiento que tengan tras detectar una alerta:

- IPS: Sistema de prevención de intrusos. Dispositivos que se encuentras online en la red y cada vez que se detecta un comportamiento anómalo mitigan el ataque bloqueando la petición.

- IDS: Sistema de detección de intrusos. Dispositivos que se encuentran offline y a priori solo detectan eventos sin realizar una mitigación del tráfico.

Según el funcionamiento de cada dispositivo de pueden diferenciar cuatro tipos de motores de detección de intrusos:

- Wireless: Detectores que monitorizan las redes Wireless disponibles, así como la actividad sospechosa detectada relacionada con los protocolos inalámbricos.

- Network Behavior Analysis: Dispositivos encargados de analizar el tráfico de la red de manera más analítica y estadística con el fin de detectar comportamientos inusuales de los flujos de la red y dispositivos que la componen.

- Network-Based: Detectores de intrusos que analizan el tráfico de la red corporativa o DMZ de la empresa.

- Host-Based: Software que se instala en un equipo terminal para realizar una protección de este. Monitoriza la actividad sospechosa trazando la actividad de los usuarios.

3.3 SIEM

Dispositivos que recolectan la información de los diferentes sistemas de seguridad. Se utilizan con el fin de correlar o cruzar los distintos tipos de información para detectar distintos tipos de ataque. La detección de ataques de realiza a través de alertas de seguridad que son desarrolladas por el administrador del SIEM[27]. Una alerta de seguridad de puede definir como un suceso en principio malicioso que ha saltado tras detectar cualquier tipo de violación, ya sea debido a las políticas de seguridad impuestas por la empresa o un intento de intrusión.

Los SIEM también pueden ser usados con fines de almacenamiento masivo, por ejemplo, tipo logger, con el fin de guardar las trazas de los sistemas para poder realizar un análisis forense sobre los datos disponibles.

El SIEM es una herramienta que monitoriza diversos datos de entrada y cuya función principal es la monitorización de seguridad de los entornos que recolecta, de este modo aporta un nivel superior de inteligencia respecto al resto de dispositivos, debido a que se pueden correlar eventos cruzados de diversas tecnologías para detectar un ataque o vulnerabilidad.

3.4 WAF

Un WAF[2] es un firewall de aplicaciones web que supervisa, filtra o bloquea el tráfico HTTP hacia una aplicación web. La diferencia entre un firewall y un WAF es que, un firewall protege tráfico entre los servidores, mientras que un WAF protege páginas web contra ataques como SQLi, XSS etc.

Los WAFs se implementan frente a las aplicaciones web, analizan el tráfico basado en la web, detectando y bloqueando tráfico malicioso. Aplican un conjunto de reglas en las peticiones HTTP para proteger las aplicaciones de ataques comunes. Pueden proteger una aplicación web específica o un conjunto de aplicaciones web.

Los WAFs se sitúan entre la aplicación web y el servidor final, de este modo se protege el origen intentando evitar DDoS, ejecución de scripts en los servidores, inyecciones de código pudiendo conseguir datos confidenciales.

4 Tipos de Ataque

4.1 Proyecto OWASP

Para realizar pruebas de intrusión se utiliza una metodología concreta que se basa en revisar las tareas con el fin de encontrar posibles fallos de seguridad. Una de las metodologías más conocidas para el desarrollo web es OWASP[24].

OWASP es el acrónimo que se traduce como Proyecto Abierto de Seguridad de Aplicaciones Web. Proyecto con código abierto con el fin de determinar y combatir:

- Los ataques web más utilizados
- Fallos de administradores de servidores web
- Fallos cometidos por programadores en el desarrollo de aplicativos.

OWASP es un organismo sin ánimo de lucro que se encarga de generar documentación y crear herramientas para la revisión servidores y aplicativos. El fin de ello es correlar todos los tipos de ataques más comunes y vulnerabilidad es actuales.

Presenta un informe que se publica cada cierto tiempo, actualizando así la variación de vulnerabilidades. El último fue publicado a principios de 2017 y el anterior fue publicado en el 2013. Se puede observar la distinción entre ambos.

OWASP Top 10 - 2013		OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Figura 4-1: Comparativa OWASP 2013 – OWASP 2017

La metodología OWASP será la usada para las pruebas de intrusión de este proyecto, aunque será una versión más reducida. Se ha adaptado el diseño de su metodología a las necesidades del proyecto.

4.2 Top 10 OWASP

Una vulnerabilidad en los sistemas de información se define como una debilidad de todo tipo que un atacante puede utilizar para comprometer la seguridad de un sistema informático. Se podrían dividir las vulnerabilidades en tres grupos:

- Vulnerabilidades de diseño: Vulnerabilidades relaciones con diseños erróneos de protocolos de red, malas políticas de seguridad y arquitecturas de red.

- Vulnerabilidades de implantación: Vulnerabilidades relacionadas con errores de programación de aplicativos, implementaciones mal desarrolladas de fabricantes, protocolos...

- Vulnerabilidades de uso: Vulnerabilidades que se dan mediante la explotación de un sistema informático, es decir, una mala configuración en un aplicativo expuesto al exterior con más datos de los necesarios.

Se detallan el top 10 de vulnerabilidades definidas por OWASP[1]:

4.2.1 Injection

Inyecciones de código tales como SQL, EL, OGNL, OS o LDAP, ocurren cuando datos malintencionados son enviados a un sistema como parte de una consulta. Las peticiones no son legítimas, intentado dañar al sistema que lo recibe, provocando así que devuelva información confidencial o no autorizada para hacerse pública.

La revisión del código fuente es el mejor método para detectar si las aplicaciones son vulnerables a las inyecciones, además de realizar pruebas automáticas exhaustas de todos los parámetros, encabezados, URL, cookies, JSON, SOA y entradas de datos XML.

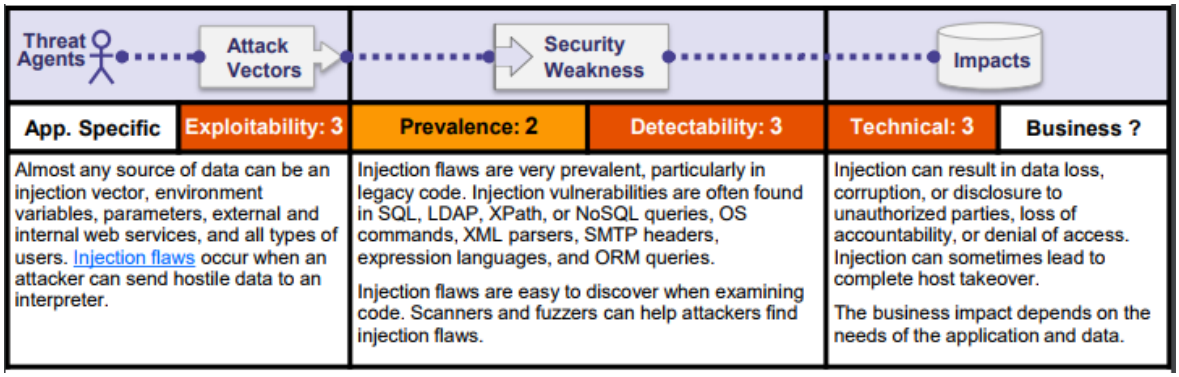


Figura 4-2: Diagrama informe OWASP A1-Injection

4.2.2 Broken Authentication

Sistemas de autenticación de usuarios, tal como login o registro de un nuevo usuario, son un foco de vulnerabilidad debido a una mala implementación. Permite al atacante obtener las credenciales, datos bancarios, ID de sesiones etc., llevando a la suplantación de usuarios.

Esto permite ataques de fuerza bruta y otros ataques automatizados. El uso de contraseñas predeterminadas o débiles permite al usuario entrar de manera autorizada, pudiendo conseguir nombres de usuario y contraseñas validadas, además, también es importante tener una recuperación de las credenciales de usuario eficaz. Las ID de sesión es un foco importante, no se debe exponer al público, y debe rotar tras el inicio de sesión, además si

no se valida correctamente durante el cierre de sesión o en el periodo de inactividad se pueden quedar abiertas de manera prolongada.

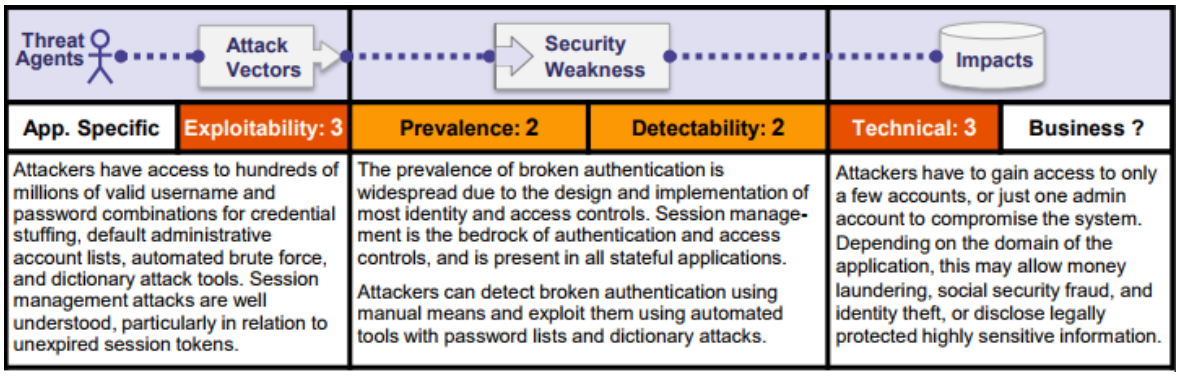


Figura 4-3: Diagrama informe OWASP A2-Broken Authentication

4.2.3 Sensitive Data Exposure

La exposición de datos sensibles puede no tener la protección adecuada, como, por ejemplo, números de cuenta bancarios almacenados sin cifrar, contraseñas enviadas en claro etc. La exposición de estos datos sin medidas de seguridad facilita al atacante obtener todo lo que busca.

Se debe determinar las necesidades de protección de los datos en tránsito y almacenados, como contraseñas, número de tarjetas de crédito, registros de salud, información personal y secretos comerciales empresa-cliente que están bajo las leyes de privacidad. Las empresas deben revisar, entre otros, todos los siguientes puntos:

- ¿Los datos confidenciales están almacenados en texto claro, incluidas copias de seguridad?
- ¿Hay algoritmos criptográficos débiles?
- ¿El agente de usuario verifica que el certificado del servidor sea válido?
- ¿Se están usando métodos claves de criptografía por defecto, sin un rotado o repetidas?

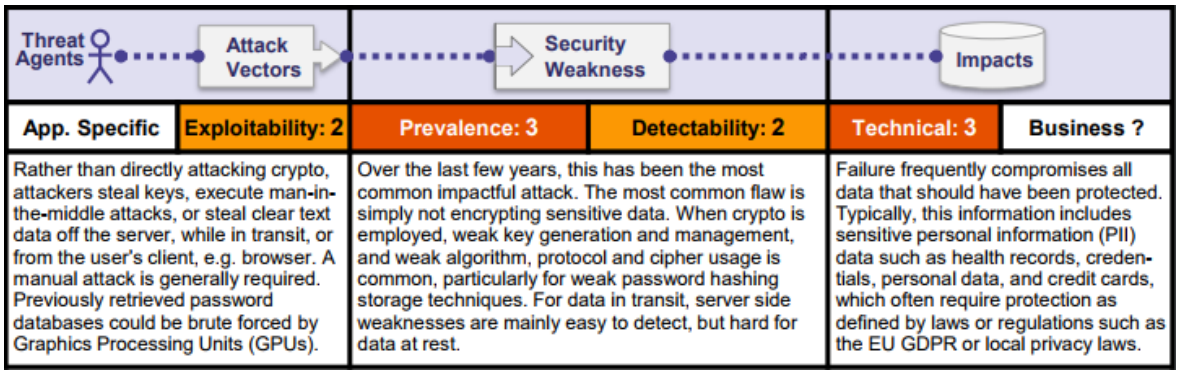


Figura 4-4: Diagrama informe OWASP A3-Sensitive Data Exposure

4.2.4 XML External Entities

Los atacantes pueden explotar vulnerabilidades XML, pudiendo ejecutar código XML directamente o cargar documentos con el fin de ejecutarlos. Esto permite acceder a ficheros privados de los sistemas, como por ejemplo extrayendo datos de los servidores en /etc/passwd o intentos de denegación de servicio provocando un bucle infinito en algún fichero.

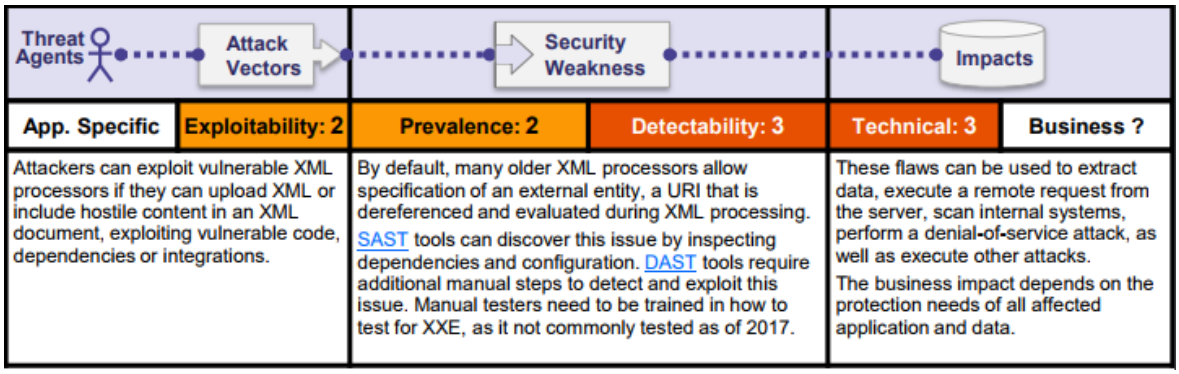


Figura 4-5: Diagrama informe OWASP A4-XML External Entities

4.2.5 Broken Access Control

Vulnerabilidad provocada por no aplicar medidas de seguridad en los usuarios de un sistema. Se deben aplicar restricciones sobre los permisos de los usuarios, debido a que, si no, esto provoca que un atacante pueda acceder a las funcionalidades e información para la que no tiene autorización.

El control de acceso aplica la política de modo que los usuarios no puedan actuar fuera de los permisos previstos. Los fallos llevan a la divulgación, alteración o destrucción de información no autorizada. Este tipo de vulnerabilidades incluyen:

- Omitir las verificaciones de control de acceso modificando la URL, a la página HMTL o un ataque mediante la API.
- Elevación de privilegios, haciendo un mal uso del usuario administrador.
- Manipulación de metadatos, como por ejemplo una cookie, un campo oculto un token de control de acceso JSON.

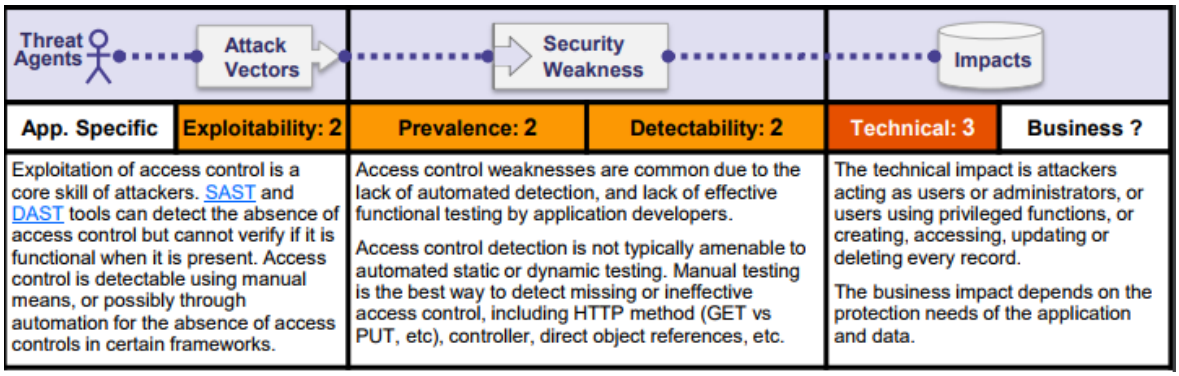


Figura 4-6: Diagrama informe OWASP A5-Broken Access Control

4.2.6 Security Misconfiguration

Fallos de configuración de seguridad que permiten al atacante obtener información y acceder a las funciones de los sistemas. Los administradores de los equipos deben revisar de manera continua las configuraciones que ellos mismos ponen, y todas aquellas que vienen por defecto. Además, es imprescindible que mantenga el software actualizado, de modo que todos los parches de seguridad se apliquen, y así no dejar huecos de seguridad abiertos.

Se debe tener un control de todas las funciones que están habilitadas e instaladas, como puertos abiertos, servicios levantados, páginas abiertas al público etc. Además de no dejar ningún tipo de configuración, contraseñas…con las predeterminadas.

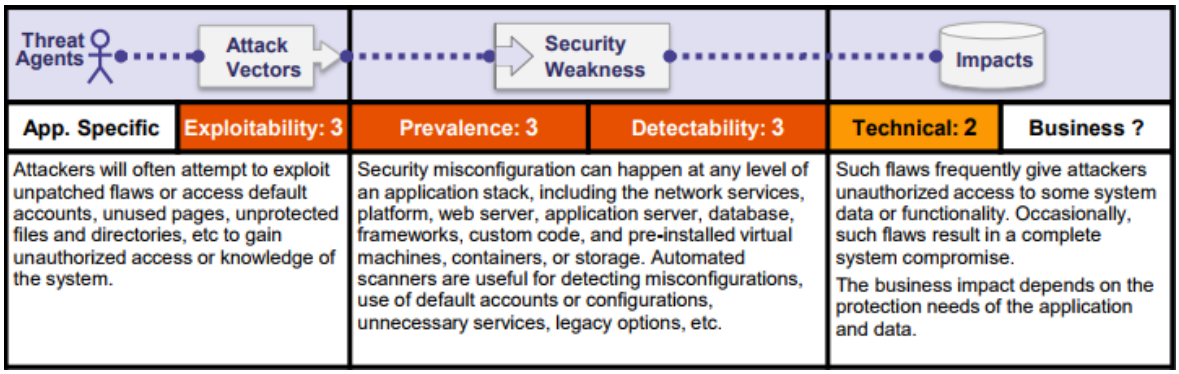


Figura 4-7: Diagrama informe OWASP A6-Security Misconfiguration

4.2.7 Cross-Site Scripting (XSS)

Vulnerabilidad basada en aplicaciones encargadas de tomar los datos del usuario y los envían a la aplicación web sin ser validados previamente, permitiendo al atacante introducir todo tipo de comandos que son ejecutados en el servidor origen sin autorización.

Estos ataques permiten que el atacante ejecute HTML arbitrario y JavaScript en el navegador del usuario, teniendo que interactuar con un enlace malicioso que apunta a una página específica del atacante, web maliciosa, anuncios o similar.

Los ataques XSS más comunes son aquellos que incluyen robo de sesión, toma del control de una cuenta, MFA bypass, ataques contra el navegador del usuario etc.

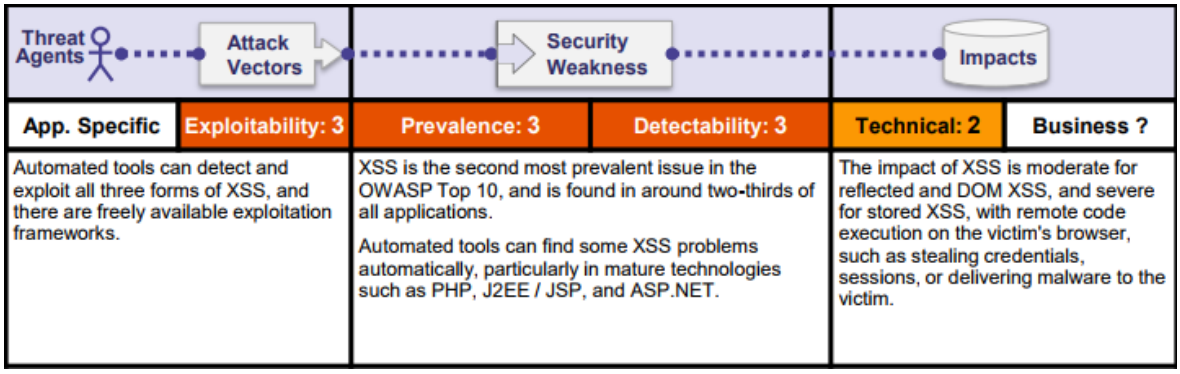


Figura 4-8: Diagrama informe OWASP A7-Cross-Site Scripting (XSS)

4.2.8 Insecure Deserialization

Las aplicaciones y las APIs son un foco de vulnerabilidades si sus hosts son deserializados. Esto provoca ataques hacia la estructura de objetos y datos donde el atacante modifica la lógica de la aplicación o logra ejecutar código remoto para cambiar el comportamiento de los sistemas antes o después de la deserialización. También pueden provocar ataques de manipulación de datos típicas, tomando el control de acceso pudiendo cambiar todo el contenido interno.

- La serialización puede darse lugar en varios tipos de aplicaciones:
- Protocolos de servicios web
- Almacenamiento de caché
- Bases de datos, servidores de caché, sistemas de archivos
- Cookies HTTP, parámetros de formularios HTML
- Comunicación remota e interproceso

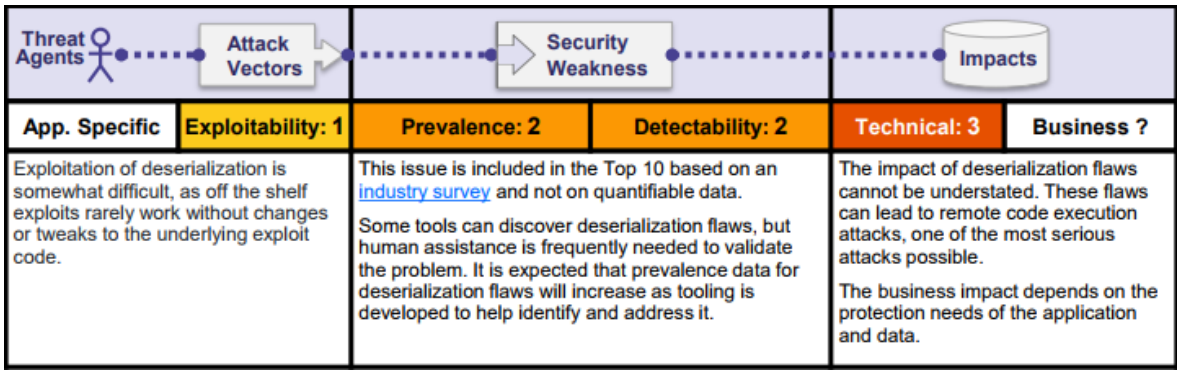


Figura 4-9: Diagrama informe OWASP A8-Insecure Deserialization

4.2.9 Using Components with Known Vulnerabilities

El uso de componentes con vulnerabilidades conocidas aumenta el riesgo de sufrir un ataque, por lo que es esencial mantener todos los sistemas y librerías actualizados. Las debilidades de los componentes es información pública accesible para todo el mundo, por lo que pueden explotar la vulnerabilidad en cualquier momento, de modo que en cuanto salga un parche para cubrirlo, se deberán poner en el sistema.

Para evitar estos ataques es necesario conocer las versiones de todos los componentes que se usan, además de saber en la duración de vida de un parche, mucho pueden ser temporales. Los desarrolladores de software deben comprobar la compatibilidad de las actualizaciones y las bibliotecas usadas.

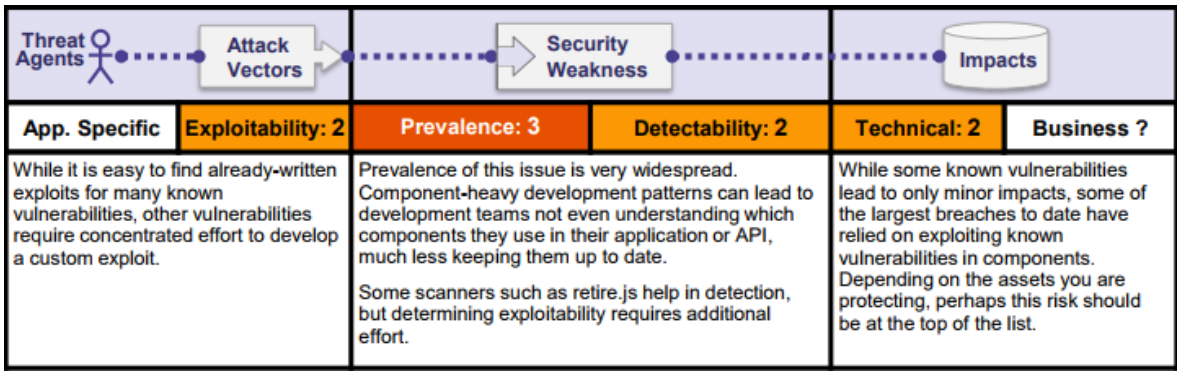


Figura 4-10: Diagrama informe OWASP A9-Using Components with Known Vulnerabilities

4.2.10 Insufficient Logging & Monitoring

Este tipo de vulnerabilidad se da cuando no se tiene una fuga de información, es decir, cuando no todos los eventos son monitorizados, provocando no tener control sobre lo ocurrido en los sistemas. Eventos como inicio de sesión o inicios de sesión fallidos no registrados, advertencias o errores no claras o inadecuadas, registro de aplicaciones no monitorizadas en busca de actividad sospechosa, almacenamiento local, umbrales de alertas no apropiados etc.

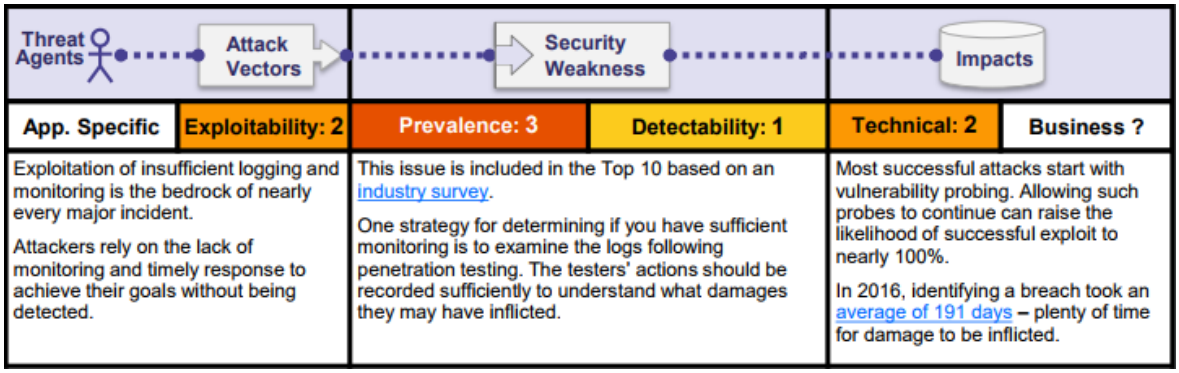


Figura 4-11: Diagrama informe OWASP A10-Insufficient Logging & Monitoring

4.3 Principales conclusiones tras OWASP

Tras analizar todo lo que la organización OWASP muestra, dejan claro que siguiendo unas líneas básicas de seguridad se consigue reducir de manera drástica el riesgo de que un equipo de la red se vea comprometido.

Recapitulando, lo más importante es:

- Mantener una comunicación abierta entre el equipo de desarrollo y el equipo de seguridad, de modo que se implemente código seguro.
- Mantener todos sistemas actualizados, teniendo un control de las versiones y qué es lo que se tiene instalados/abierto/levantado.
- Validar todo datos introducido por el usuario antes que llevar al servidor final.
- Proteger todos los servidores finales de modo que no lleguen peticiones ilegítimas.

5 Opciones de Protección para Empresas

Para decidir que arquitectura implementar y la ubicación de cada herramienta de protección es necesario tener en cuenta los siguientes puntos:

- ¿Cuáles son las herramientas de protección más extendidas?
- ¿Qué es lo que busca la empresa para poder adaptar las medidas de protección a su objetivo?
- ¿Es necesaria una implementación avanzada?

Las herramientas más extendidas son los firewalls, seguido de los detectores de intrusos (IDS/IPS). Una vez que las herramientas han llegado a su madurez es conveniente incluir un SIEM en la arquitectura, con el fin de almacenar de manera masiva todos los eventos del sistema, pudiendo así generar alertas y realizar análisis forense.

Este es el grado de madurez al que suelen llegar las empresas y entidades públicas actualmente.

5.1 Cortafuegos (Firewall)

.
A la hora de elegir el firewall[25] es necesario saber que implementa cada tecnología en un único dispositivo, de modo que se puede comparar las funcionalidades que tiene cada uno y así adaptarlo más a las necesidades del cliente y a los costes. Funcionalidades que son importantes a tener cuenta:

- Funcionalidad de cortafuegos.
- Funcionalidad de acceso remoto VPN.
- Funcionalidad de proxy, obligando a que todas las máquinas de la empresa salgan a Internet a través del firewall canalizando las conexiones a través del proxy.
- Funcionalidad de envío de trazas a un syslog. Esto es necesario para poder posteriormente integrar el Firewall en un correlador, el cual recibirá todos los eventos de firewall, procesándolo y generando alertas.
- Funcionalidad antivirus. Necesario para el análisis de los paquetes en busca de virus, de este modo, se tiene controlado la posible propagación de virus.
- Funcionalidad IPS. Control del tráfico de paquetes dentro de la empresa.
- Configuración de reglas por parte del administrador del equipo.
- Control de aplicaciones. Controlando así el uso de aplicaciones web dentro de los trabajadores de la empresa, pudiendo restringir el acceso redes sociales, webs pornográficas etc, todo aquello no corporativo.

5.1.1 Palo Alto

Palo Alto[23] es una empresa que tiene desarrolladas diversas herramientas de seguridad de pago. Entre ellas tiene varios dispositivos que actúan como firewall. Antes solo existían dos opciones básicas dentro de sus firewalls: bloquearlo todo para mantener la seguridad de la red o habilitarlo todo para fortalecer las actividades empresariales. Actualmente, acaban de sacar una nueva generación más sofisticada.

Inspecciona todo el tráfico, incluidas las aplicaciones, amenazas y el contenido, vinculándolo a los usuarios independientemente de la ubicación o dispositivo. Permite conciliar la seguridad con las iniciativas empresariales.

Una comparativa de cómo han avanzado los dispositivos:

PA-5020 frente a PA-5220

Rendimiento y capacidad*	PA-5020	PA-5220
Rendimiento del cortafuegos (App-ID habilitado)	5 Gbps	18 Gbps
Rendimiento de Threat Prevention	2 Gbps	9 Gbps
Rendimiento de VPN IPSec	2 Gbps	5 Gbps
Nuevas sesiones por segundo	120 000	169 000
Número máximo de sesiones	1 000 000	4 000 000
Sistemas virtuales (de base/máximos)**	10/20	10/20

PA-5050 frente a PA-5250

Rendimiento y capacidad*	PA-5050	PA-5250
Rendimiento del cortafuegos (App-ID habilitado)	10 Gbps	35 Gbps
Rendimiento de Threat Prevention	5 Gbps	20 Gbps
Rendimiento de VPN IPSec	4 Gbps	14 Gbps
Nuevas sesiones por segundo	120 000	348 000
Número máximo de sesiones	2 000 000	8 000 000
Sistemas virtuales (de base/máximos)**	25/125	25/125

PA-5060 frente a PA-5260

Rendimiento y capacidad*	PA-5060	PA-5260
Rendimiento del cortafuegos (App-ID habilitado)	20 Gbps	72 Gbps
Rendimiento de Threat Prevention	10 Gbps	30 Gbps
Rendimiento de VPN IPSec	4 Gbps	21 Gbps
Nuevas sesiones por segundo	120 000	458 000
Número máximo de sesiones	4 000 000	32 000 000
Sistemas virtuales (de base/máximos)**	25/225	25/225

Figura 5-1: Nueva generación dispositivos Palo Alto

Principales propiedades de la última generación de Palo Alto:

- Identifica los hosts infectados por bots y las interrupciones de la actividad en la red a causa del malware.
- Limita las trasferencias de archivos y datos no autorizadas.
- Controla la navegación web.
- Aplica políticas basadas en dispositivos para el acceso a aplicaciones.
- Confirma automáticamente los hosts que se encuentran en riesgo.
- Elaboración de informes y creación de logs.
- Plataformas virtuales o de hardware para fines específicos.
- Bloquea el malware desconocido o selectivo con WildFire.
- Previene amenazas conocidas mediante IPS y antivirus o antispyware de red.
- Un conocimiento integral del contexto equivale a políticas de seguridad más estrictas.
- Agiliza la gestión de dispositivos, redes y políticas con funciones de administración intuitivas que se adaptan a la estructura de la organización.
- Uso de la seguridad para impulsar la empresa.

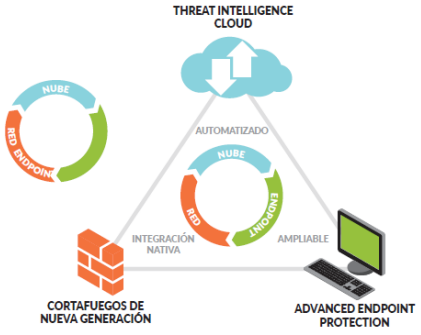


Figura 1: Plataforma de seguridad de nueva generación de Palo Alto Networks

Figura 5-2: Plataforma de seguridad de nueva generación de Palo Alto Networks

5.1.2 Software libre – Endian Firewall

Endian Firewall[22] es una herramienta de software libre. Es un firewall UTM que se instala en las empresas y entidades, el cual cumple una gran cantidad de funciones. Por tanto, este firewall se puede considerar entre lo que es un cortafuegos clásico y lo que actualmente se considera un "NG Firewall" (Next Generation Firewall), es decir, un firewall que contiene el control de aplicaciones, IPS y antivirus en un mismo dispositivo.

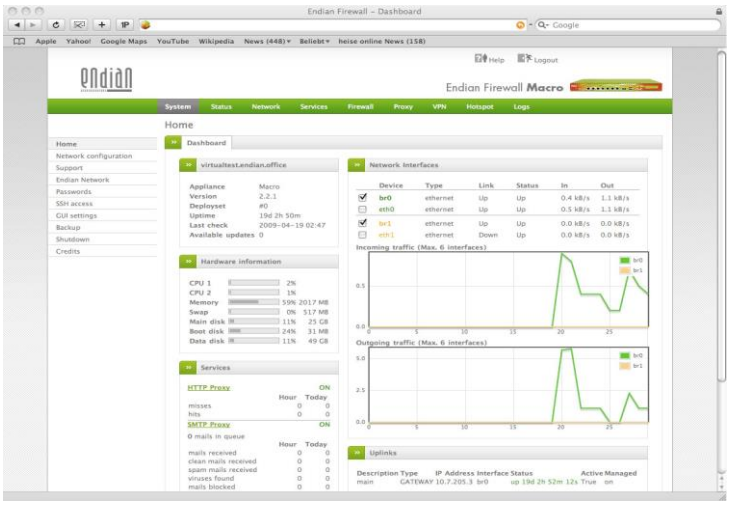


Figura 5-3: Interfaz gráfica Endian Firewall

Principales propiedades:

- Funcionalidad de cortafuegos.
- Funcionalidad de acceso remoto VPN.
- Interfaz de usuario para la configuración de las reglas. (Esta parte facilita mucho la administración del dispositivo frente al habitual firewall de tipo "Iptables").
- Funcionalidad de proxy. De esta manera se obliga en a que todas las máquinas salgan a Internet a través de este firewall canalizando las conexiones a través del proxy.
- Funcionalidad de envío de trazas a syslog remoto. Esta parte ha sido también bastante importante ya que de esta manera se integra con el correlador de logs, el cual recibe los eventos del firewall y los procesa para generar alertas.

5.2 IDS/IPS

Se debe saber cómo se quiere enfocar la prevención de intrusos, es decir, si se desea solo detectar o denegar los ataques que se encuentren[26].

Estas herramientas se basan en el análisis del tráfico de red, compara firmas de ataques conocidos o comportamientos sospechosos, como escaneos de puertos, paquetes malformados, accesos permitidos después de un escaneo de red etc. Normalmente se integra con un firewall, debido a que es un puerto donde forzosamente deben pasar los paquetes y pueden ser bloqueados antes de penetrar en la red.

Disponen de una base de datos de firmas con ataques conocidos, para así poder detectar anomalías.

El IDS, es decir, sistema de detección de intrusos, suele tener sensores virtuales con los que el núcleo del IDS puede obtener datos externos. El IDS detecta gracias a esos sensores, las anomalías que pueden ser indicio de la presencia de ataques y falsas alarmas.

El IPS de red, es decir, un sistema de prevención de intrusos basados en la detección de ataques a través de análisis de tráfico de red que pasan a través de él. Dispositivo de seguridad de red que monitoriza el tráfico de red y/o las actividades de un sistema, en busca de actividad maliciosa. Entre sus principales funciones, se encuentran no sólo la de identificar la actividad maliciosa, sino la de intentar detener esta actividad.

5.2.1 McAfee

McAfee[4] es una empresa que tiene dos herramientas desarrolladas. Una para prevenir intrusiones de nueva generación pudiendo bloquear las amenazas y otra como antivirus.

Propiedades principales de la herramienta para prevenir intrusiones:

- Bloquea intrusiones: Detiene nuevos ataques y desconocidos con una inspección basada en firmas y sin firmas. La tecnología de detección de intrusiones sin firmas permite al IPS identificar el tráfico de red malicioso y detienen los ataques desconocidos para los que no existen firmas.
- Seguridad física y de la nueva unificada: La compatibilidad con VMware NSX y OpenStack permite a las organizaciones unificar la seguridad informática en todas las redes físicas y virtuales.
- Seguridad y rendimiento: Plataforma de hardware que alcanza velocidades de más de 320 Gbps.
- Se adapta a amenazas a tiempo real: La compatibilidad con McAfee Threat Intelligence Exchange proporciona conocimiento de las amenazas en tiempo real en redes tanto físicas como virtuales. La integración con McAfee Advanced Threat Defense y McAfee MOVE AntiVirus permite a las organizaciones automatizar la seguridad avanzada para centros de datos definidos por software.

5.2.2 Software libre – Suricata

Suricata[21] es una herramienta de software libre. Evolución multihilo de snort, el cual es uno de los IDS más desplegados y utilizados en el mundo de software libre. Existe una gran comunidad de usuarios que utilizan snort y diseñan nuevas firmas para detectar los nuevos patrones de ataques a medida que aparecen nuevas vulnerabilidades.

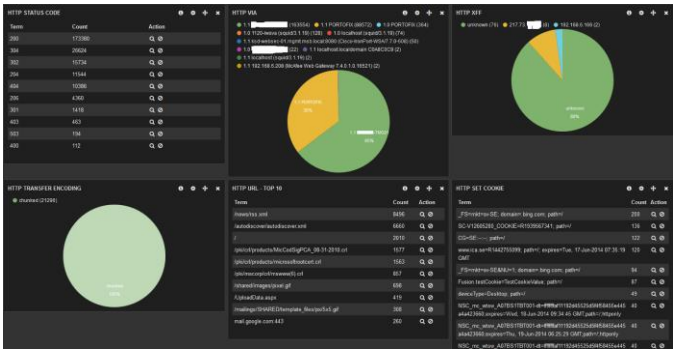


Figura 5-4: Interfaz gráfica Suricata

En esta herramienta se pueden cargar distintos paquetes de reglas profesionales, con el fin de que se apliquen sus firmas al tráfico. Hay paquetes de más de 12000 firmas, relacionadas con la detección de ptrones tanto de ataques clásicos, como los ataques más nuevos. Estos son paquetes de pago que cubren muchas vulnerabilidades, incrementando así la fiabilidad de las detecciones y bloqueos que se realizan. Sus firmas están diseñadas por analistas de seguridad.
Esta herramienta puede configurarse en distintos modos, IDS, IPS network security monitoring (NSM) y offline pcap processing.

5.3 SIEM

El SIEM[27] elegido debe tener unas cualidades mínimas para una correcta correlación:

- Capacidad de correlación de eventos en tiempo real. En caso de no ser a tiempo real, no advertirá de eventos maliciosos en el momento de un ataque y puede ser demasiado tarde para actuar.
- Capacidad de almacenamiento masivo, de este modo se pueden analizar los datos en busca de patrones, evidencias, análisis forense, extracción de datos para reportes.
- Presentación normalizada de los datos independientemente de la tecnología. Es importante que los eventos se parseen correctamente, debido a que es la única manera de generar alertas y de que el administrador del equipo pueda interpretar los datos que se están almacenando. Las tecnologías no tienen por qué tener un parseador de eventos para todos los eventos que se integren, sin embargo, el administrador podrá crear sus propios parseadores para que esto no sea una limitación.
- Comunidad de soporte para los administradores, de modo que tengan de manera eficiente ayuda de primera mano de los fabricantes. Esto es algo sumamente valorado.
- Capacidad de crear reglas de correlación de una manera intuitiva y sencilla.

5.3.1 ArcSight

ArcSight[5] es una herramienta de pago. Una tecnología que permite realizar análisis en tiempo real de las alertas de seguridad generadas por el hardware y las aplicaciones de red. Posibilita la transformación de grandes volúmenes de datos en inteligencia de seguridad procesable. Se pueden integrar todo tipo de equipos para monitorizar sus eventos, desde Firewalls, IDS/IPS, WAFs, hasta servidores Directorios Activos para tener un registro de todo lo ocurrido con las cuentas de empleados.

Se pueden generar reports, alertas, dashboards etc todo lo necesario para poder exprimir al máximo la información que brindan los logs y poder correlar en busca de actividad ilícita. No es una herramienta para la monitorización de salud de las herramientas integradas, sin embargo, podría ser usada como tal, por ejemplo, para controlar las caídas de phase2 o tunnels de VPNs de la empresa.

Tiene tres productos disponibles:

- ArcSight Enterprise Security Manager (ESM): Analiza diferentes amenazas dentro de una base de datos y correlaciona las vulnerabilidades en función del nivel de riesgo
- ArcSight Express: Analiza las amenazas dentro de una base de datos y correlaciona las vulnerabilidades en una escala mucho más pequeña que el ESM
- ArcSight Logger: Transmite datos en tiempo real y los categoriza en registros específicos

5.3.2 Software libre - OSSIM

OSSIM[6] es una herramienta de software libre el cual tienen una integración nativa con el detector de intrusos Suricata. Detecta amenazas centralizadas y responde a incidentes en entornos de nube, infraestructura local y aplicaciones en la nube. Gestiona registros para el cumplimiento continuo e investigaciones forenses. Además, genera informes de cumplimiento precompilados para PCI DSS, HIPAA, NIST CSF y más.

Figura 5-5: Interfaz gráfica OSSIM

Sus principales cualidades son:

- Correla eventos a tiempo real
- Amacena de manera masiva todos los eventos
- Dispone de una comunidad de soporte
- Normaliza los datos independientemente de la tecnología
- Se pueden crear reglas de correlación

5.4 WAF

Un WAF[2] es un firewall de aplicaciones web que supervisa, filtra o bloquea el tráfico HTTP hacia una aplicación web. La diferencia entre un firewall y un WAF es que, un firewall protege tráfico entre los servidores, mientras que un WAF protege páginas web contra ataques como SQLi, XSS etc.

5.4.1 Akamai

Las soluciones de seguridad web basadas en la nube de Akamai[20] se han diseñado para aprovechar la potencia de Intelligent Platform para ofrecer detección, identificación y

mitigación de ataques distribuidos de denegación de servicio (DDoS) y de nivel de aplicación, a la vez que garantizan la disponibilidad y el rendimiento de sus propiedades web.
Ha gestionado varios de los mayores ataques DDoS en un pasado reciente, lo que ha permitido ahorrar millones de dólares en ingresos y proteger los datos de clientes.

Dispone de un catálogo de reglas para proteger la web controlando ataques:

- SQL Injection
- Cross Site Scripting (XSS)
- Command Injection
- Invalid HTTP
- Remote File Inclusion
- PHP Injection
- Trojan
- DDOS
- Rate Control

Tiene un módulo para controlar el tráfico BOT, pudiendo detectar si detrás de una petición hay una persona o una máquina, en base al tipo de anomalías en la petición y el número de peticiones por segundos.

Además, tienen un módulo que recalcula la reputación de una IP en base al número y tipo de ataques que realiza una misma IP. El algoritmo usado tiene la inteligencia de estar en continua evolución. Si un IP deja de tener tráfico ilegítimo, la reputación de la IP irá decreciendo progresivamente.

5.4.2 Software libre - DVWA

DVWA[7] es un WAF de software libre, es una aplicación web PHP/MySQL. El objetivo es ayudar a los administradores de una plataforma probar sus habilidades y herramientas en un entorno legal, y ayudar a los desarrolladores web a comprender los procesos de seguridad de las aplicaciones web.

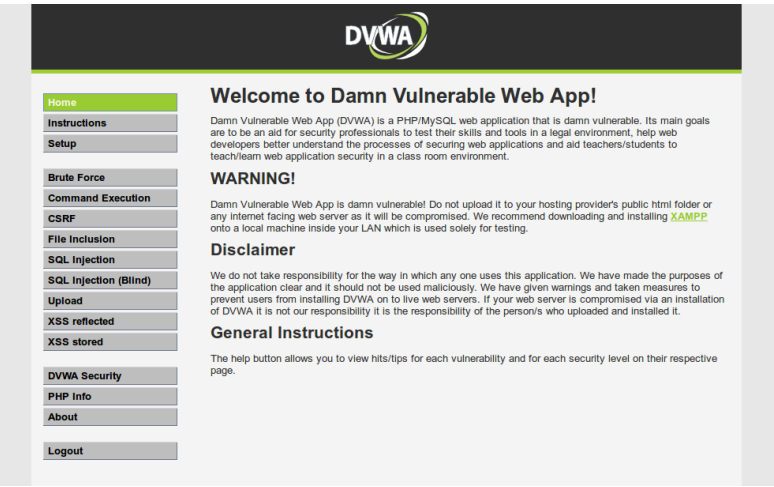


Figura 5-6: Interfaz gráfica DVWA admin

6 Intentos de Ataque

6.1 Ataques

Se detallan las principales vulnerabilidades:

6.1.1 SQL Injection

Ataques de inyección SQL explotan vulnerabilidades generados por errores de programación web. Los errores suelen darse cuando no se validan las entradas introducidas por los usuarios de las páginas web, como por ejemplos en campos para el registro de un usuario, una página de login o un campo de búsqueda.

Un atacante puede aprovechar la vulnerabilidad para inyectar código SQL adicional para alterar el funcionamiento normal de la página web, pudiendo conseguir acceder a las bases de datos que no deben estar disponibles. Dependiendo del tipo de inyección de código SQL se puede hacer más o menos daño en el entorno pudiendo descargarse la base de datos completa, accediendo a datos privados de usuarios como números de tarjetas de crédito o realizar un DROP de sus tablas, dejando a la empresa sin datos.

Ejemplo de un ataque SQLi[7]:

En un formulario que se solicita el ID de un usuario para así poder obtener el nombre y apellido del usuario. Ejecución normal:

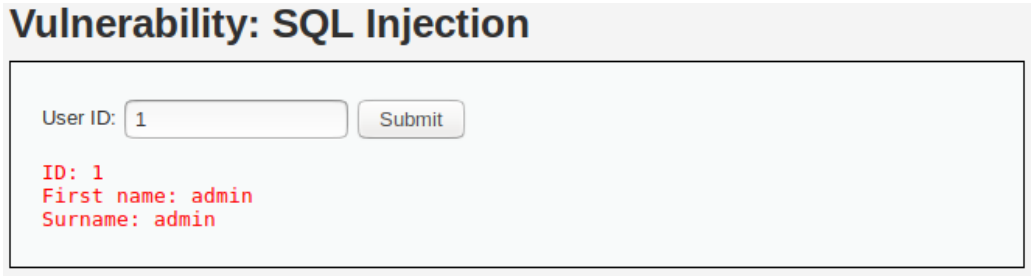


Figura 6-1: Vulnerabilidad SQL Injection – Ejecución normal

El campo debería ser validado previamente a llegar la petición al servidor, recibiendo así solo números. Sin embargo, al no verificar el valor del campo, acepta letras y cualquier otro tipo de caracteres, pudiendo así modificar la query que se ejecuta en la base de datos para conseguir el nombre y apellidos del usuario al que pertenece el ID insertado.

Se puede intuir que la query inicial deberá ser algo del estilo:

SELECT name, surname FROM users where id = '<ID>';

Por lo que, queda por poner una condición en la consulta que será evaluada como verdadera en todo momento para que saque todos los datos de la base de datos, como por ejemplo id = 1 or 1 = 1:

SELECT name, surname FROM users where id = ‘1’ or ‘1’=’1’;

Esto será evaluado como verdadero mostrando lo siguiente:

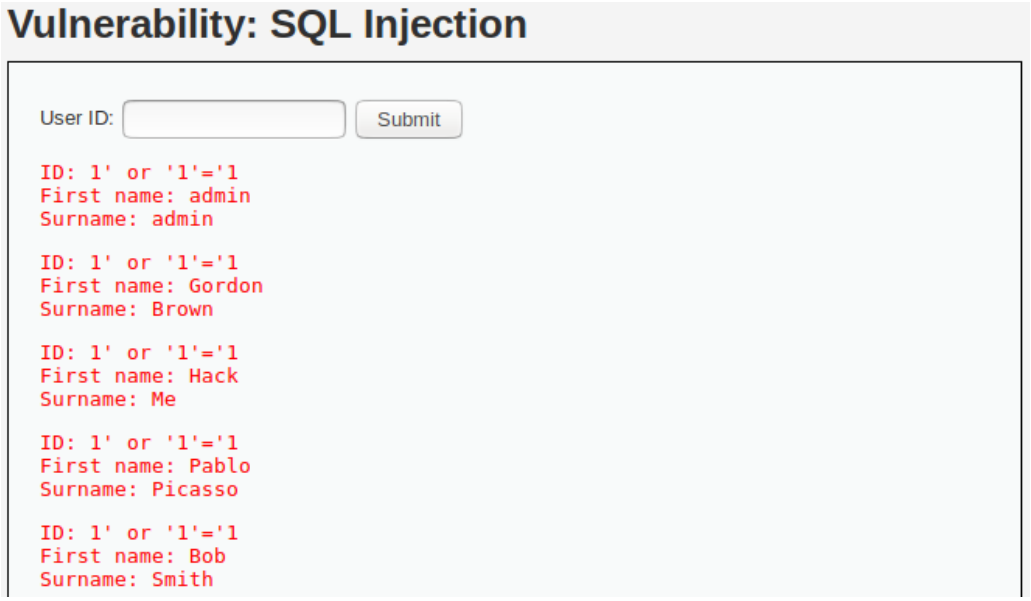


Figura 6-2: Vulnerabilidad SQL Injection – Ataque

Para evitar este tipo de ataques es imprescindible desarrollar código seguro. Ejemplo de un código PHP vulnerables a ataques SQLi:

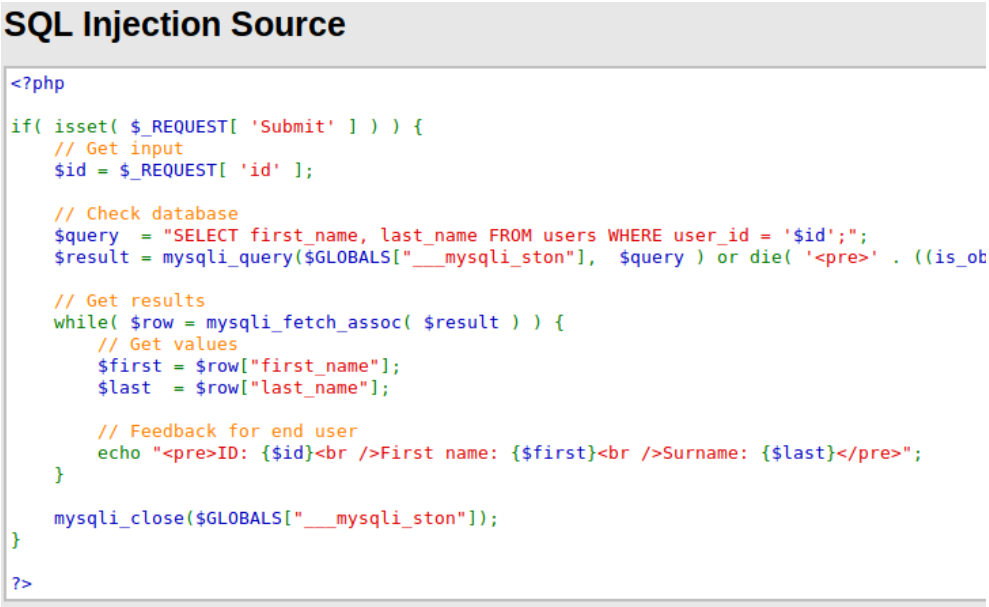


Figura 6-3: SQL Injection Source

Pasos para realizar una programación segura:

1. Escapar caracteres en los campos de entrada de datos. Todo dato introducido debe tratarse como un string, evitando así introducir comandos posteriormente ejecutados. Funciones como mysql_escape_string, evitan esta vulnerabilidad.

2. Validar los datos de entrada. Se debe comprobar todo tipo de evidencias que se puedan validar para comprobar que el tipo de dato recibido es el que se espera, es decir, que, si se espera un número, comprobar que es un Integer, si debe tener una longitud fija, comprobar que tiene esa longitud, si debe tener letras y números, comprobar que tiene ambas.
3. Tener un SIEM. Esto permitirá generar alertas de seguridad a tiempo real basadas en patrones, como, por ejemplo, controlas diversos intentos fallidos desde una misma IP y múltiples usuarios. Esto permite poder actuar frente a un ataque.

6.1.2 ShellShock

Vulnerabilidad que afecta a gran cantidad de equipos Linux y Unix. Permite mediante la definición de una variable, la ejecución de un comando con elevación de privilegios. Es una vulnerabilidad peligrosa si se dispone de servidores que utilizan la Shell de tipo BASH en servidores web que interpreten CGI de manera automática.

Esta vulnerabilidad es relativamente nueva que es fácilmente explotado.

Ejemplo de un ataque ShellShock[13]:

Bash continúa ejecutando código tras la definición de una función en una variable de entorno. Se puede definir una función de la siguiente manera:

```
’(){ echo "Funcion"; };’
```

Posteriormente se puede definir una variable de entorno Bash con un método:

```
env VARDENTORNO=’(){ :; };’
```

El fallo es que Bash no se detiene en el último punto y coma:

```
env VARDENTORNO=’(){ :; };’ echo "Se puede ejecutar lo que queramos."’
```

El echo tras el punto y coma, podría ser la ejecución de cualquier otra función, pudiendo explotar la vulnerabilidad con facilidad.

6.1.3 File Inclusion

Vulnerabilidad que permite subir un fichero a un directorio web con el fin de posteriormente ejecutarlo. Este caso se podría dar, por ejemplo, en servidores web que alojan imágenes y no tienen correctamente validada la carga de ficheros, esto permite al atacante aprovechar la vulnerabilidad para subir scripts para ejecutarlos de manera remota.

Ejemplo de File Inclusion[19]:
Este tipo de vulnerabilidad aparece cuando un php no está programado de manera segura. Código vulnerable:

```
<?php
include $_GET[‘pagina’];
?>
```

Mientras que el siguiente código no es vulnerable:

```
<?php
include('pagina.php');
?>
```

La primera es vulnerable debido a que se puede modificar los parámetros de las funciones. El GET pasa los datos por URL, por lo que en la barra de direcciones URL se podría insertar código. Por ejemplo:

http://www.pagina.com/index.php?pagina=34g4.php

Se podría modificar 34g4.php, modificando así lo que se incluye mostrar, pudiendo ejecutar otros ficheros o hacer directory transversal hasta el fichero /etc/hosts del sistema.

En la segunda no se puede modificar debido a que no hay nada que se puede modificar ya que directamente lo que hace es incluir un .php.

6.1.4 Cross Site Scripting (XSS)

Inyecciones de código con el fin de que el servidor web lo interprete y lo ejecute realizando tareas para las que no fue diseñado. Con este tipo de ataques se lleva a robar credencial o datos comprometidos de los usuarios. Este tipo de vulnerabilidad de explota con lenguajes de programación que el servidor web es capaz de interpretar como scripts, por ejemplo, JavaScript y HMTL.

Ejemplo de ataque XSS:

Página web en donde solicita un nombre para posteriormente imprimirlo por pantalla:

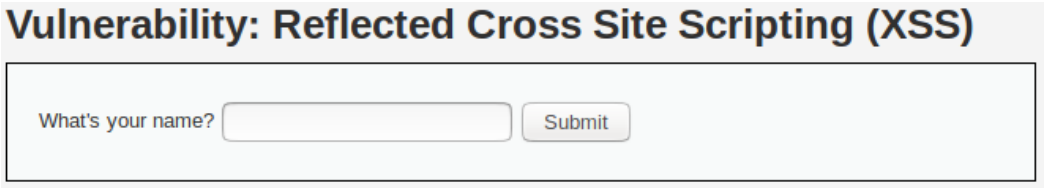


Figura 6-4: Reflected Cross Site Scripting – Ejecución normal

Si en vez de introducir nuestro nombre, se introduce código que pueda ser interpretado y ejecutado:

<script>alert('XSS Vulnerable')</script>

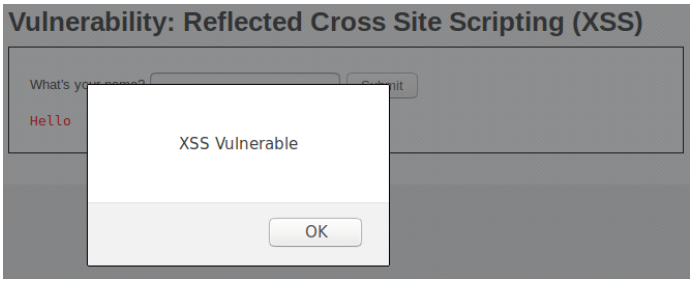


Figura 6-5: Reflected Cross Site Scripting – Ataque

Se ha ejecutado el script introducido por línea de comandos.

Para poder evitar este tipo de ataques, se deben detectar patrones en las peticiones en busca de palabras clave en los lenguajes de programación, como *script*, “<”, “>”, ““”.

6.1.5 Directory Transversal

Vulnerabilidad que permite al atacante acceder a ficheros o directorios raíz del sistema operativo que aloja un servidor web. Para poder explotar esta vulnerabilidad es necesario ejecutar un servidor web con un usuario con privilegios de administrador o un super usuario. De este modo, el atacante puede acceder a ficheros como el fichero Linux que contiene el listado de usuarios (etc/passwd) al poner en el servidor web de Apache sobre el que está montado el sistema Linux/Unix.

Ejemplo de un ataque Directory Transversal:

Al igual que en línea de comandos para subir de directorio se usa el comando *cd ..* o *cd /../*, esto puede ser usado para realizar un ataque escalando así en el directorio, llegando a archivos que no deberían estar abiertos al público.

http://web.com/../../../../../../etc/passwd

6.1.6 Inyección de Comandos

Vulnerabilidad causada por fallos de programación en el cual el servidor web interpreta el texto que un atacando puede insertar en un campo de texto libre como un campo de búsqueda. Para aprovecharse de la vulnerabilidad, inserta comando que el servidor ejecuta, forzando a este a realizar tareas para las que no está diseñado.

Ejemplo de ataque de inyección de comandos:

Campos de entrada de un servicio que realiza ping a dispositivos tras introducir una dirección IP. En esta ejecución que pide hacer un ping a sí mismo:

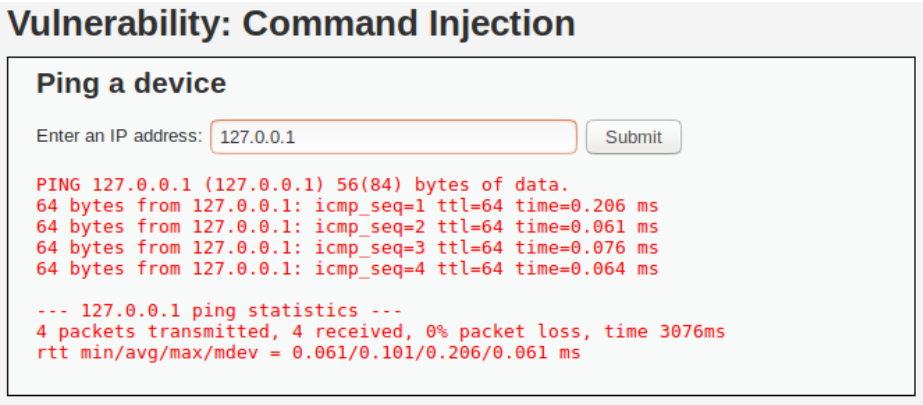


Figura 6-6: Command Injection – Ejecución normal

El campo debería ser validado previamente a realizar un ping a la dirección IP para evitar futuros ataques, sin embargo, no lo valida, por lo que se puede ejecutar un comando de

manera malintencionada, obteniendo así todos los datos del fichero /etc/passwd del servidor final:

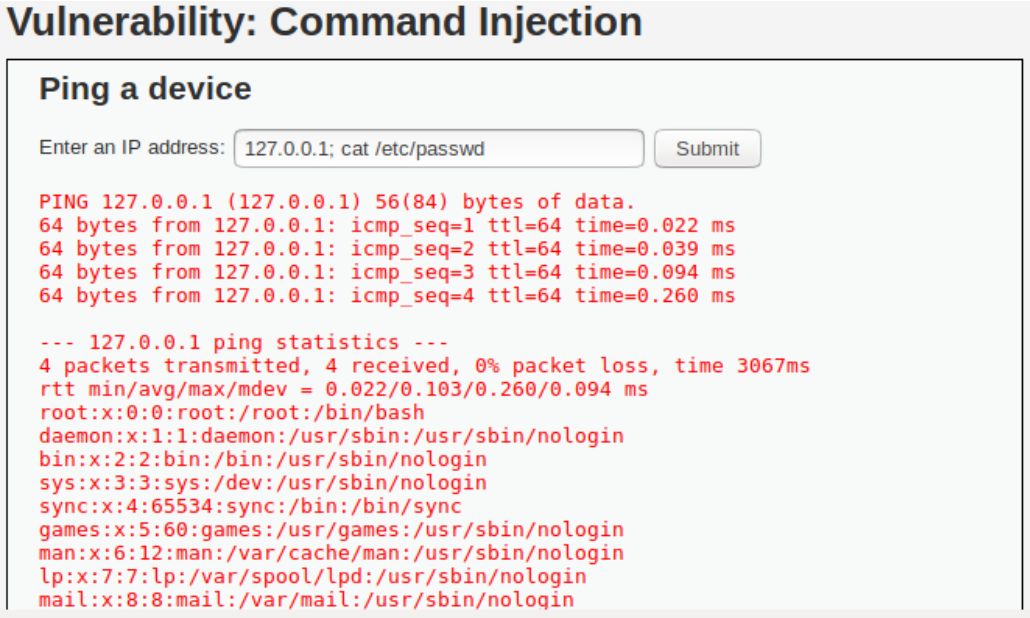


Figura 6-7: Command Injection – Ataque

6.1.7 Inyección XML

Este tipo de vulnerabilidad es muy similar a inyecciones de SQL, la diferencia es que la inyección de código está orientado a bases de datos XML en vez de SQL. Vulnerabilidad provocada por la no validación de la entrada de datos procedentes de usuarios, formularios de login, registro de nuevos usuarios, campos de búsqueda etc. Los datos se insertan a través de la página web, utilizando el error con el fin de alterar las consultas que realizan en XM, obteniendo así información de la empresa y pudiendo realizar daños persistentes.

Ejemplo de Inyección XML[18]:

En la autenticación web de un usuario para logarse se usa un XML:

```
<?xml version="1.0" encoding="utf-8"?>
<Empleado>
  < Empleado ID="1">
    <Nombre>Cilene</ Nombre >
    <Apellido>Arroyo</ Apellido >
    <Usuario>CA</ Usuario >
    <Password>pass</Password>
</ Empleado >
```

Si el usuario introduce:

Usuario: qwer’ or 1=1 or ‘a’=’a
Password: qasd

Esto se traduce en:

Empleado[Usuario/text()='qwer' or 1=1 or 'a'='a' And Password/text()='qasd']

Que de manera lógica es equivalente a:

[(Usuario/text()='qwer' or 1=1) or ('a'='a' And Password/text()='qasd')]

Esto permite que con que el usuario sea true funcione, independientemente de la contraseña, ya que 1=1 se cumplirá.

6.1.8 Fuerza Bruta

Este tipo de ataque realiza comprobaciones masivas de autenticación con una gran cantidad de usuarios y contraseñas, con esto se intenta conseguir acceder al sistema. Estos ataques explotan fallos de configuración en los servidores, provocado por aplicaciones que no están bien configuradas. Por ejemplo:

- No hay una limitación en el número de conexiones e intentos de autenticación de usuarios
- No existen medidas de seguridad, por ejemplo, bloqueando temporalmente cuentas de usuarios o IPs para el acceso a servidores.
- No está limitada la cantidad de autenticaciones en un periodo de tiempo.
- Las políticas de seguridad de contraseñas de usuario no son lo suficientemente estrictas.

Ejemplo de ataque de fuerza bruta:

Se puede crear un script el cual se ejecutará para realizar ataques de fuerza bruta en una autenticación de una página web. Los intentos de autenticación se realizan a través de peticiones web enviando el usuario y la contraseña en la URI de la petición. El conjunto de usuarios y contraseñas con los cuales el script intenta realizar la autenticación puede ser diverso, de este modo, generará múltiples fallos de autenticación en las plataformas.

Para que sea un ataque más o menos agresivo, se deberá indicar el intervalo de tiempo con el que se debe ejecutar recursivamente el script, por ejemplo, si se indica que pase medio segundo entre cada petición no será muy agresivo, pero suficiente para ser detectado como alto automático.

7 Defensa ante Vulnerabilidades del Entorno

En este proyecto se va a describir un ejemplo de esquema de red viable para implementar en una empresa como seguridad informática. Se indicarán las características principales tanto a nivel hardware como software de cada una de las máquinas que se han utilizado. Se ha trabajado con VMWare Workstation Pro[17] para las simulaciones del entorno. Se han usado tecnologías open source.

Se ha montado un WAF y un SIEM debido a que el objetivo de este proyecto es centrarse en montar un firewall de aplicaciones web, sin embargo, en un esquema de seguridad de este tipo, también sería conveniente montar un firewall perimetral como, por ejemplo, un pfSense que es código abierto, por lo que se mostrarán las reglas y la configuración que tendría el firewall.

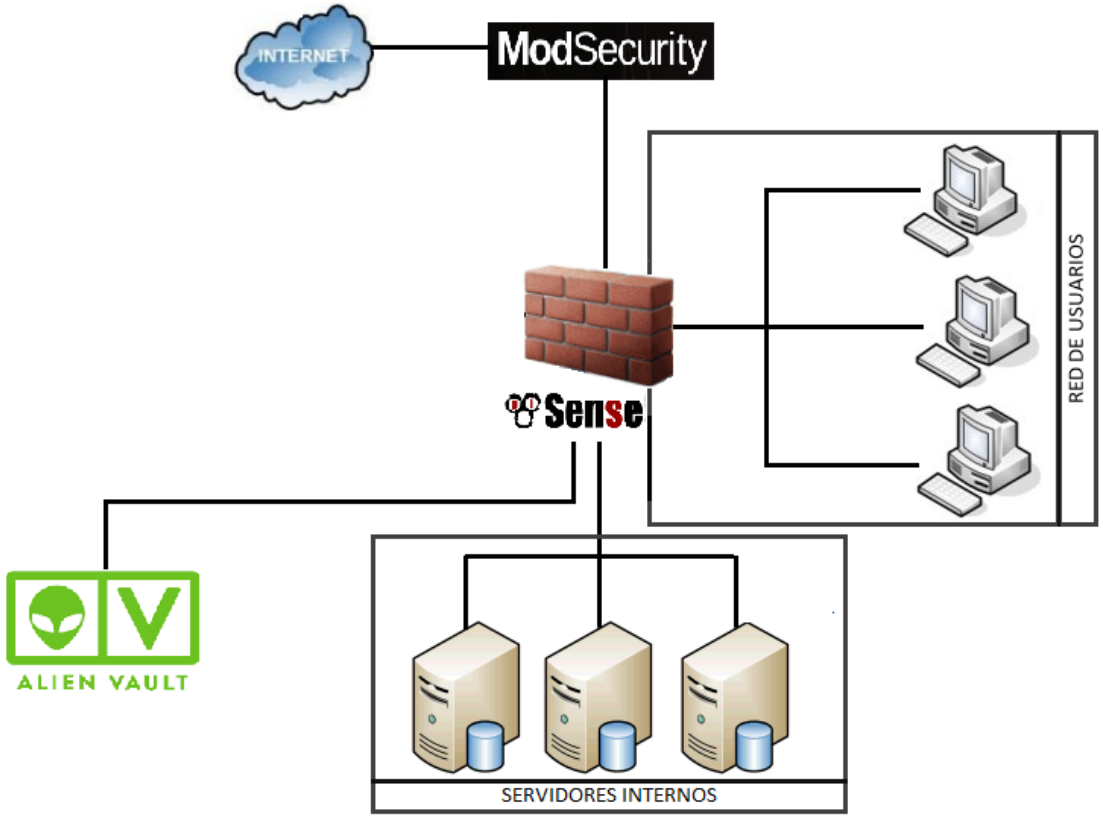


Figura 7-1: Esquema de red

7.1 WAF – ModSecurity

ModSecurity[15] es un WAF open source que soporta Apache HTTP, Microsoft IIS y Nginx.

Para la instalación de ModSecurity[15] se ha montado una Máquina Virtual Debian con las siguientes características:

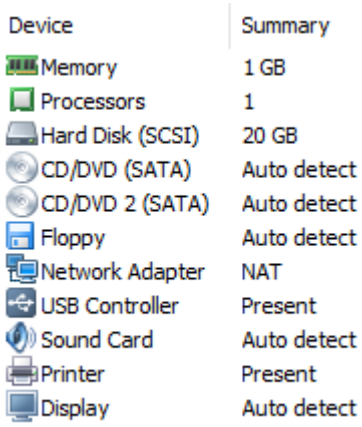


Figura 7-2: MV ModSecurity

Por consola se han ejecutado los siguientes comandos:

$ sudo apt-get install libapache2-mod-security
$ sudo a2enmod mod-security
$ sudo /etc/init.d/apache2 force-reload

Las reglas que vienen integradas en los WAFs controlan los ataques que se exponen entre otros, teniendo alertas que controlan las peticiones que machean con las siguientes anomalías:

- XSS: Búsqueda en el contenido de la request HTTP como, por ejemplo:
  - '<font','<form', '<frameset','<h1','<head','<html'
  - 'background', 'href', 'lowsrc', 'src, 'dynsrc'

- Trojan: Búsqueda en el contenido de la request HTTP como, por ejemplo:
  - Cabeceas como x_key
  - 'text/plain', 'text/html

- SQLi: Búsqueda de contenido en la request HTTP como, por ejemplo:
  - 'sec_to_time', 'soundex' o 'system_user'
  - ‘drop’, ‘select’, from’, ‘unique’, ‘union’, ‘select’
  - ~#!@$*()%^&
  - ‘case’, ‘join’, ‘like’, ‘or’, ‘and’, ‘<’, ‘>’

- Comon web attacks

- Malicious activity

- Local File Inclusion: Búsqueda de contenidos en la request HTTP como, por ejemplo:
  - /etc/passwd
  - php.ini
  - my.conf

- Remote File Inclusion: Búsqueda de contenidos en la request HTTP como, por ejemplo:
    - URL que acaban en '?
    - Peticiones a páginas que contienen http, https y ftp
    - Intentos de acceder a ficheros sensibles del sistema como: htpasswd o htgroup

- Remote Code Execution

- PHP Code Injection: Búsqueda de contenidos en la request HTTP como, por ejemplo:
    - '<?'
    - 'safe_mode', 'open_basedir', 'disable_functions'
    - 'preg_last_error(', 'base64_encode(', 'zlib_decode('

- HTTP Protocol Violations: Peticiones sin Accept Header, sin User-Agent, sin cookie session etc.

- Shellshock: Peticiones HTTP que contienen el string "(){" en las request_cookies, request_headers, request_protocol, request method etc.

Las reglas que vienen con ModSecurity sería aconsejable profundizar en la creación de firmas específicas para el entorno. Consiguiendo así aumentar la tasa de detección y bloqueo de la plataforma sin añadir costes a la infraestructura.

7.2 Firewall - pfSense

Usar una tecnología como pfSense[16] es una buena opción para una empresa con recursos limitados. Las características que podría tener el firewall:

Memoria: 512 MB
Procesadores: 1
Disco Duro: 2GB

La configuración más sencilla y básica sería permitir únicamente el tráfico por el puerto 443 y puerto 80, al resto del tráfico se le aplicaría un drop.

El firewall perimetral podría configurarse con tres interfaces. La primera haría de puerta con la red WAN, siendo así la salida con el exterior. La segunda interfaz sería un adaptador LAN que se comunicaría con el tráfico interno. Se podría mandar una copia de todo el tráfico al SIEM de modo que se integrarían todos logs para correlarlos. La tercera interfaz sería la red de gestión, siendo la IP usada por los administradores de red para gestionar el firewall.

Se configurarían las siguientes reglas básicas. Políticas que definiría si un paquete se aceptará o será tráfico denegado. Estas reglas se basan en la IP, el protocolo y el puerto origen y destino.

ID	Acción	Protocolo	IP Origen	Puerto Origen	IP Destino	Puerto Destino
1	ACCEPT	UDP	*	>1023	*	80
2	ACCEPT	UDP	*	>1023	*	80,443
	DENY	*	*	*	*	*

Tabla 7-1: Reglas básicas firewall

Una configuración un poco más robusta sería controlar el puerto 53 (DNS) y el puerto 22.

ID	Acción	Protocolo	IP Origen	Puerto Origen	IP Destino	Puerto Destino
1	ACCEPT	TCP	*	>1023	*	80
2	ACCEPT	TCP	10.0.0.0/24	>1023	*	80,443
3	ACCEPT	UDP	*	>1023	*	53
4	ACCEPT	UDP	*	53	*	>1023
5	ACCEPT	UDP	*	>1023	*	22
	DENY	*	*	*	*	*

Tabla 7-2: Reglas firewall

Una empresa tiene zonas que administrar, como la red de usuarios, servidores internos etc, esa configuración sería algo más compleja y basada en la arquitectura de red del cliente.

7.3 SIEM – OSSIM

Se ha montado una Máquina Virtual que contendrá el SIEM open source de AlienVault, denominado como OSSIM[6]. Características de la máquina:

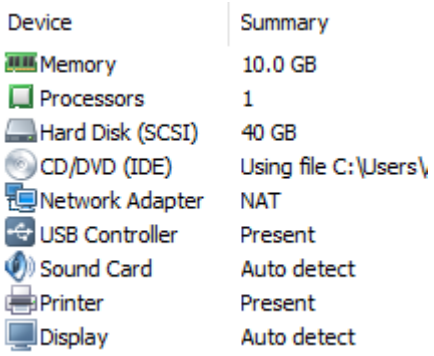


Figura 7-3: MV OSSIM

La instalación del SIEM se ha detallado en el Anexo B. El OSSIM es una herramienta útil para la detección de intrusiones y prevenir los ataques. Permite crear alertas de correlación en base a los distintos eventos recibidos de las herramientas integradas.

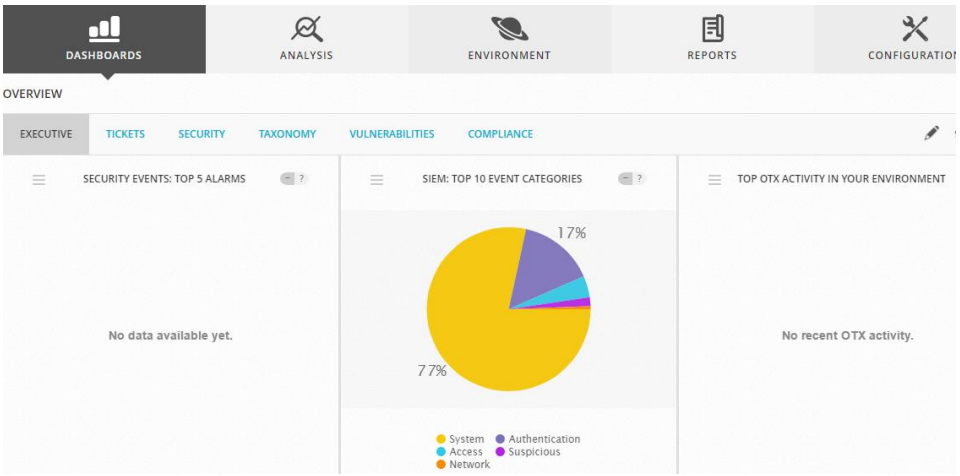


Figura 7-4: Interfaz gráfico OSSIM

Se deberá dotas a la plataforma de una inteligencia mediante los logs de las fuentes integradas correlando los eventos. Para poder realizarlo, es necesario diseñar reglas de correlación avanzadas.

Unos ejemplos de las reglas que podrían diseñas para detectar anomalías:

Integración en el SIEM exclusivamente del WAF:

Denominadas como críticas:

- Alto número de alertas desde una misma IP origen en un periodo de tiempo. Por ejemplo, 500 alertas de SQLi y PHPi desde una misma IP origen en 5 minutos.
- Alto número de alertas denegadas por el WAF desde una misma IP origen. Por ejemplo, 200 alertas XSS denegadas en el WAF desde una misma IP origen.
- Alto número de alertas en una misma URL destino. Por ejemplo, 300 alertas PHPi, SQLi y Remote File Inclusion hacia una misma URL.

Alertas para el control diario del tráfico:

- Múltiples alertas WAF en 5 minutos desde una misma IP. Por ejemplo, 60 alertas de Shellshock en 5 minutos desde una misma IP origen.
- Múltiples alertas WAF en 10 minutos hacia una misma URL. Por ejemplo, 30 alertas de intentos de Brute forcé hacia una misma URL.

Es conveniente realizar una lista de URLs sensibles en concordancia con la empresa para poder seguridad extra en páginas de login, registro, reservas, compras etc, de este modo se alertarán intentos de denegaciones de servicio distribuidos y se evitarán perdidas de dinero a la empresa por la caída del servicio.

Integración en el SIEM del WAF y firewall perimetral:

- Detección de una IP que genera varios eventos contra varias máquinas destino distintas.
- Escaneo de IPs en un puerto concreto desde una IP seguido de un acceso conseguido.
- Escaneo vertical de puertos desde una IP interna hacia una IP interna seguido de un acceso correcto.

- Múltiples escaneos desde una IP interna.
- Posible propagación de malware: Varios equipos infectados con un mismo malware (si el firewall tiene modo antivirus)
- Posible ataque de DOS desde una IP interna contra otra IP interna

Con estas reglas de correlación, se deberá aplicar una política de seguridad que bloquee temporalmente la IP del atacante cuando machee con algunas de las reglas. El bloqueo se podría definir a distintos niveles, a nivel WAF o a nivel de firewall perimetral.

8 Conclusiones y trabajo futuro

8.1 Conclusiones

Con la evolución de Internet es muy importante tener en mente la seguridad de datos en la red. Cuando se es una empresa, es quizá más importante, debido a que se manejan datos de terceros que deben ser confidenciales.

Cuando se tiene claro que es algo imprescindible, es algo que debe llevarse a cabo con coherencia, siendo conscientes de hasta donde se quiere llegar y cuáles son los objetivos a corto/medio plazo. Se debe tener coherencia debido a que, no es de gran utilidad comprar una tecnología muy cara, implantarla y no realizar un mantenimiento de ella, pretendiendo que con solo ponerla va a hacer todo el trabajo sola.

Existen muchas tecnologías en el mercado que son grandes herramientas muy potentes, sin embargo, debe haber un humano por detrás poniendo la lógica. La tecnología está en una continua evolución, por lo que se debe estar al día de las vulnerabilidades y los parches que salen para remediarlos.

Siempre se debe tener en cuenta que tener una seguridad absoluta no es viable, siempre hay una nueva forma de vulnerar el sistema, sin embargo, se tiene una amplia visión de los ataques más recibidos en las empresas y sin fácilmente de mitigar siempre y cuando se cumplan una serie de requisitos como los siguientes:

- Sistemas y librerías actualizadas.
- Desarrollo seguro.
- Aplicación de parches en sistemas y librerías.
- Tener sistemas de detección de intrusiones como IPS, IDS o/y WAF.
- Monitorización de todos los eventos de las tecnologías de la empresa para poder correlarlos.
- Dar de baja a cuentas de usuarios que no deberían tener acceso al sistema.
- Buena gestión de las políticas de las herramientas.
- Realizar backups periódicos en caso de fallo.

Las medidas básicas previenen los riesgos más altos de sufrir un ataque, el sistema se hará más robusto cuando se creen reglas y configuraciones más específicas para el tipo de mercado de la empresa y las necesidades de la empresa. Además, es esencial que el equipo de seguridad tenga comunicación con la mayoría de departamentos, para poder amoldar la seguridad en todos los entornos.

8.2 Trabajo futuro

Una vez implantando lo básico en el proyecto, en este caso un WAF y un SIEM, el proyecto debe seguir creciendo. La seguridad informática puede llegar hasta los niveles más bajos dentro de la empresa, llegando a controlar hasta quién accede al proxy de las oficinas hasta qué aplicaciones están siendo ejecutadas en los pcs de cada empleado.

Los siguientes pasos tras lo expuesto serían:

- Implantación de firewalls, tanto perimetrales como firewalls internos
- Implantar un antivirus como Symantec o McAfee para tener un control del malware dentro de la red
- Implantar un IDS/IPS para tener un control del tráfico dentro de la red
- Si la empresa tiene sedes, como, por ejemplo, grandes tiendas, centros, hoteles…etc. Convendría poner un Firewall en cada una de las redes, así se pondrá una capa de seguridad y posteriormente integrar los FW en el SIEM y correlar los eventos.
- En el caso de ser necesario tener más Firewalls al mismo nivel, poner un balanceador entre ambos como F5 para que la carga se distribuya de manera coherente.
- Tener una configuración de proxy común en todas sus sedes para poder correlarlo y buscar anomalías.
- Integración de los Directorios Activos para tener un control del uso de las cuentas administrador (Intentos de ataque desde dentro de la organización) o intentos de conseguir usuarios y sus contraseñas mediante un ataque Brute Force.

Toda tecnología que se configure es recomendable integrar sus logs en el SIEM. Es la única manera de correlar eventos, por ejemplo, un equipo interno genera una alerta de un escaneo vertical dentro de la red y a su vez salta una alerta de virus en ese mismo equipo. Esto indica una infección en un equipo que hace que se escaneen puertos abiertos.

El mantenimiento de cada tecnología lleva un intenso trabajo que se debe revisar diariamente debido a que la seguridad está en continua evolución. Esto llevará a la creación de un equipo técnico, que con la madurez del proyecto se acaba dando un servicio 24x7 al cliente, ya sea presencial, o en horas fuera del horario laboral, una rotación de la guardia entre los miembros del equipo. Esto consiste en tener ciertas alertas como críticas por las cuales se les llamará por teléfono para atenderlas inmediatamente.

Referencias

[1] TOP 10 OWASP, https://www.owasp.org/index.php/Top_10-2017_Top_10
[2] WAF, https://es.wikipedia.org/wiki/Web_application_firewall
[3] Sistema de detección de intrusos, https://es.wikipedia.org/wiki/Sistema_de_detecci%C3%B3n_de_intrusos
[4] McAfee, https://www.mcafee.com/es/products/network-security-platform.aspx
[5] ArcSight, https://en.wikipedia.org/wiki/ArcSight
[6] AlienVault, https://www.alienvault.com/products/ossim
[7] DVWA, http://www.dvwa.co.uk/
[8] Virus informáticos, https://smarterworkspaces.kyocera.es/blog/8-tipos-virus-informaticos-debes-conocer/
[9] Historia de la seguridad, http://wwwmantovaniseguridadinformatica.blogspot.com.es/2009/08/un-poco-de-historia.html
[10] Historia de la seguridad, https://blogs.technet.microsoft.com/ponicke/2007/04/19/seguridad-informatica-un-poco-de-historia/
[11] Historia de la seguridad, https://prezi.com/vnbaj88nuq0p/historia-de-la-seguridad-informatica/
[12] Top ataques de la historia, https://www.economiadigital.es/tecnologia-y-tendencias/los-diez-mayores-ataques-informaticos-de-2016_188964_102.html#
[13] Explotación ataque Shellshock, http://blog.elevenpaths.com/2014/09/shellshock-como-se-podria-explotar-en.html
[14] Plugin AlientVault, https://www.aldeid.com/wiki/Write-AlienVault-Plugins
[15] ModSecurity, https://www.modsecurity.org/
[16] pfSense, https://www.pfsense.org/
[17] VMWare Workstation Pro, https://my.vmware.com/en/web/vmware/info/slug/desktop_end_user_computing/vmware_workstation_pro/14_0
[18] Inyección XML, https://www.owasp.org/index.php/Inyecci%C3%B3n_XPath
[19] File Inclusion, https://www.welivesecurity.com/la-es/2015/01/12/como-funciona-vulnerabilidad-local-file-inclusion/
[20] Akamai, https://www.akamai.com/es/es/products/web-performance/web-performance-optimization.jsp
[21] Suricata, https://suricata-ids.org/
[22] Endian firewall, https://www.endian.com/products/utm/
[23] Palo Alto, https://www.paloaltonetworks.com/products
[24] OWASP, https://www.owasp.org/index.php/Main_Page
[25] Firewall, https://www.google.es/search?q=firewall&oq=firewall&aqs=chrome..69i57j69i65l2j69i61j0l2.1615j0j4&sourceid=chrome&ie=UTF-8
[26] IDS/IPS, https://www.pandasecurity.com/spain/support/card?id=31463
[27] SIEM, https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=10&ved=0ahUKEwien9eA7NnYAhXmD8AKHe44BsoQFghcMAk&url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FSecurity_information_and_event_management&usg=AOvVaw1p4-wF_JczE-Z9JiX0YcTA

[28] Top ataques de la historia, https://es.gizmodo.com/los-10-mayores-ataques-informaticos-de-la-historia-1580249145
[29] Top ataques de la historia, https://www.minutouno.com/notas/1511875-cuales-fueron-los-ataques-informaticos-mas-importantes-la-historia

Glosario

API	Application Programming Interface
FW	Firewall
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
SIEM	Security information and event management
OSSIM	Open Source Security Information Management
OWASP	Open Web Application Security Project
VLAN	Virtual Local Area Network
XSS	Cross-Site Scripting
HTML	Hypertext Markup Language
IP	Internet Protocol
IPS	Intrusion Prevention System
SQL	Structured Query Language
SSH	Secure Shell
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
UDP	User Datagram Protocol
WAF	Web application firewall
XML	Extensible Markup Language

Anexos

A Posibles estructuras de proyectos para empresas

Previamente a montar un sistema de seguridad en una empresa, es necesario saber la infraestructura de la empresa y saber cómo está compuesta la red del cliente y sus necesidades.

Opción 1 – Firewall perimetral

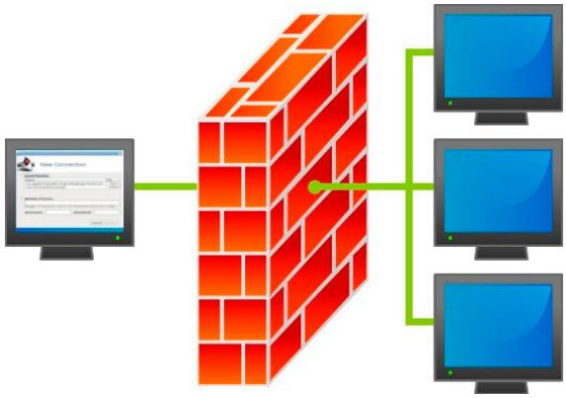


Figura A-1: Firewall perimetral siendo toda una red plana.

El beneficio de este tipo de estructura es se tiene un único punto de fallo. La configuración del Firewall debe ser más robusta y compleja debido a que va ser la única capa de seguridad. Se deberán segmentar todas las zonas desde un único punto, sin embargo, si se tienen múltiples firewalls a mismos o distintos niveles, se deberá tener un control más amplio y configurar diversas redes para que todo se complemente, teniendo más posibilidad de tener mayores puntos de fallo.

Opción 2 – Varios firewalls

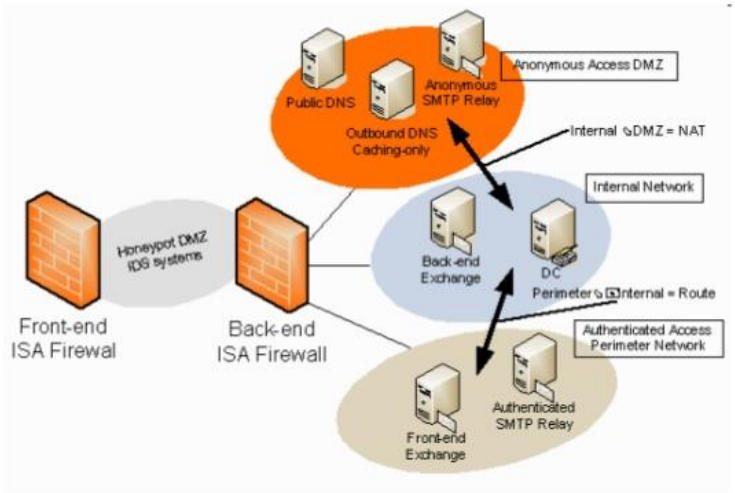


Figura A-2: Varios Firewalls segmentando la red.

Los beneficios de tener varios firewalls implantados, es que al no ser una red plana y teniendo una sed segmentada, se mejoran los problemas de saturación en red y se seguridad. Por ejemplo, si alguien lanza un broadcast para buscar un equipo, el equipo que lo lanza estará llegando a todos los equipos dentro de la red, en el caso de que sea una red grande, llegar a todos los equipos puede saturar la red.

Además, esta estructura permite tener una buena segmentación de red, esto es muy útil debido a que no es conveniente que se pueda llegar a todos los puntos de la red desde cualquier otro punto, es necesario acotar, sobre todo por seguridad. No tiene sentido que un usuario pueda alcanzar ciertos equipos de servidores internos, ya que eso debería estar segmentado en otra zona.

Es conveniente segregar, separando las zonas de la DMZ, la red de usuarios, los servidores internos etc y que cada zona tenga permiso de acceso solo a lo que le corresponda

Opción 3 – IPS en modo Inline

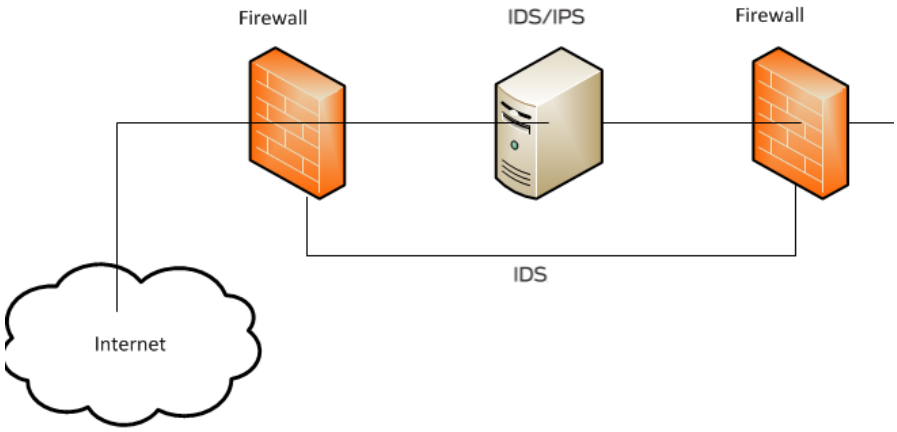


Figura A-3: IPS/IDS

Otra alternativa podría ser poner un IDS o IPS entre dos Firewalls.

Modo offline: La tecnología se configuraría en modo IDS. El primer firewall tendría una pata al segundo firewall y una segunda pata al IDS mandando así una copia de su tráfico. El IDS al estar fuera de la línea del cable que une los dos firewalls, no cortaría el tráfico en ningún momento, es decir, si el IDS se cae no cortaría el tráfico. La parte negativa es que, si una regla machea con un paquete ilegítimo, no cortaría su tráfico. Sin embargo, una manera de solventarlo sería, generar una alerta cuando machea con una regla, generando así un trigger que provocaría un bloqueo automático en los firewalls, añadiendo la IP a la lista negra del firewall y del IDS, bloqueando así todo el tráfico de esa IP, pero no tendríamos el bloqueo por paquetes como en el caso de una configuración inline.

Modo inline: La tecnología se configuraría en modo IPS. Los dos firewalls se conectarían por un solo cable, poniendo en medio de los dos un IPS. Esto provocaría que todo el tráfico pasase por el IPS estando así en bypass. El tráfico entraría por el puerto delantero y saldría por el puerto trasero. La parte negativa de este tipo de configuración es que en el caso de que el IPS se cayera el tráfico se cortaría, no llegaría al segundo firewall. La ventaja de esta configuración sería que en el momento en el que se generase una alerta provocada por

machear con una regla, como, por ejemplo, un drop, el IPS cortaría el paquete. En caso de que el paquete no machease con ninguna regla, el IPS dejaría pasar el paquete. Un IPS no generaría el bloqueo de todo el tráfico de IP como un IDS, si no que bloquearía el tráfico por paquetes, según vaya macheando con reglas.

Con la integración de un IPS/IDS se podrían implementar reglas de correlación avanzadas unidas al tráfico del firewall en un SIEM. Por ejemplo:

- Detección de un ataque desde una IP origen que ha generado 3 eventos distintos en el IPS
- Detección cuando un atacante realiza un escaneo de red y posteriormente genera un evento en el IDS
- Detección de un atacante cuando genera una cantidad muy elevada de eventos en el IPS.
- Detección de un atacante cuando genera un evento en el IPS y posteriormente en el WAF.

Opción 4 – VPN entre sedes de la empresa

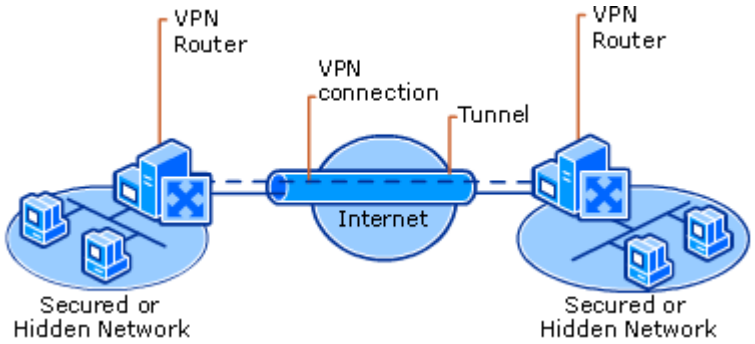


Figura A-4: VPN entre sedes

En el caso de ser una empresa con diversas sedes estando en varios países/ciudades, sería aconsejable que la comunicación entre las sedes sea mediante una VPN creando una gran red. Serían sedes diferentes compartiendo un mismo rango de IPs, siendo transparente la localización de por ejemplo un usuario, para poder así definir en un único punto todas las reglas para red de usuarios.

Esta opción es complementaria con cualquier otra estructura expuesta en el proyecto.

Con la integración de VPNs se podrían implementar reglas de correlación avanzadas en un SIEM. El objetivo de un SIEM no es correlar eventos de salud de las herramientas, sin embargo, en ese caso, los logs que generan las VPN en los firewalls son de gran ayuda para asegurarnos de que ninguna sede quede aislada por la caída de alguna VPN.

Opción 5 – Proxy

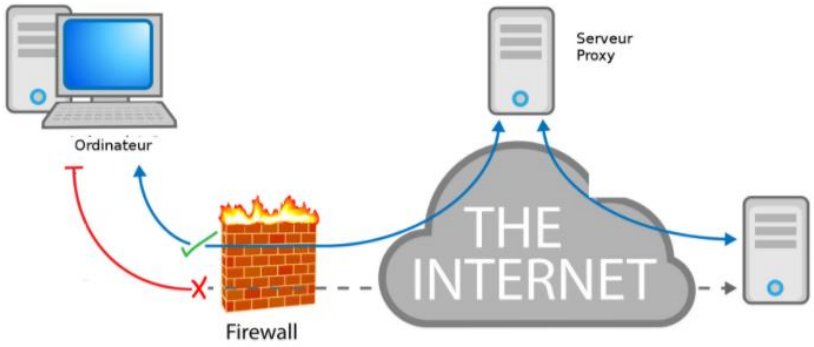


Figura A-5: Proxy interno

Poner un proxy dentro de la red, como, por ejemplo, Blouecoat.

Se podría poner un proxy delante de la red de usuarios teniendo así la autenticación de un usuario. Esto sería una ventaja debido a que en vez de tener las IPs por DHCP, intentando averiguar la IP que estuvo asignada en un periodo de tiempo en un equipo, directamente con el proxy se obtiene el usuario que tiene que logarse para poder navegar estando autenticando pudiendo ver su tráfico de manera única y directa.

Se obligaría a que todas las máquinas salieran a Internet a través de este firewall canalizando las conexiones a través del proxy.

Esta opción es complementaria con cualquier otra estructura expuesta en el proyecto.

Con la integración de un Proxy se podrían implementar reglas de correlación avanzadas unidas al tráfico del firewall en un SIEM. Por ejemplo:

- Descarga de .exe o .jar
- Bluecoat descarga sospechosa
- Detección de intento de salto de proxy
- Usuario de navegación compartido entre varios equipos
- Usuario accediendo de forma concurrente a contenido bloqueado. Categorías acordadas con el cliente que deben ser bloqueadas.
- Usuario accediendo a categorías peligrosas. Las categorías peligrosas deberán ser indicadas por la empresa, indicando que consideran como sitios homologados.

B Instalación OSSIM

Seleccionar que se quiere instalar un SIEM:

Figura B-1: Instalador

Selección del idioma:

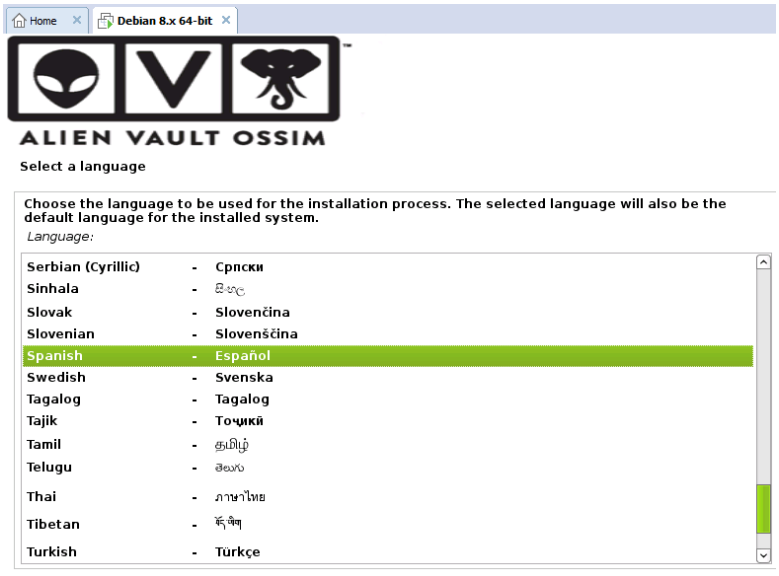


Figura B-2: Idioma

Selección de la ubicación:

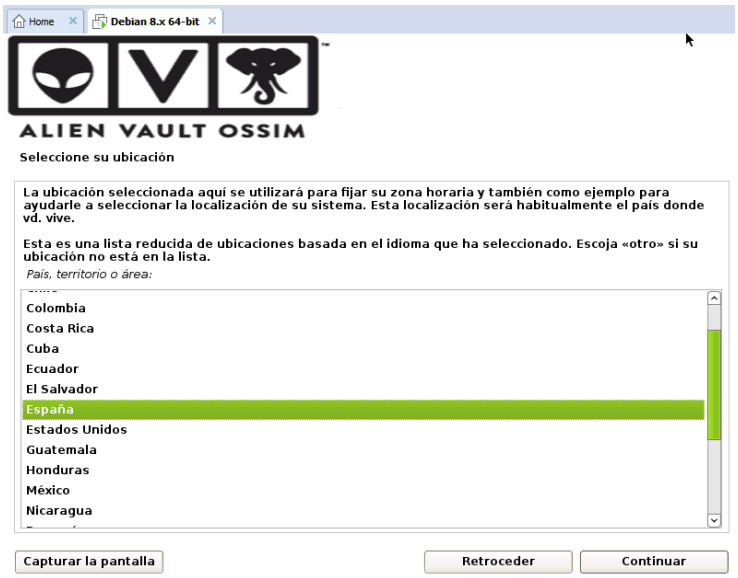


Figura B-3: Ubicación

Selección del idioma del teclado:

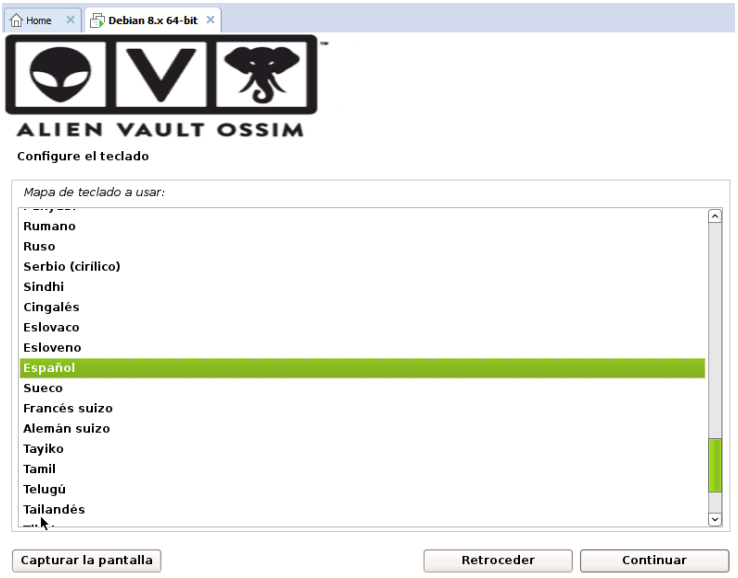


Figura B-4: Teclado

Se cargan los componente del instalador:

Figura B-5: Instalador de componentes

Configuración de la IP de la máquina:

Figura B-6: IP de la Máquina

Configuración de la máscara de red:

Figura B-7: Máscara de Red

Configuración del encaminador de red:

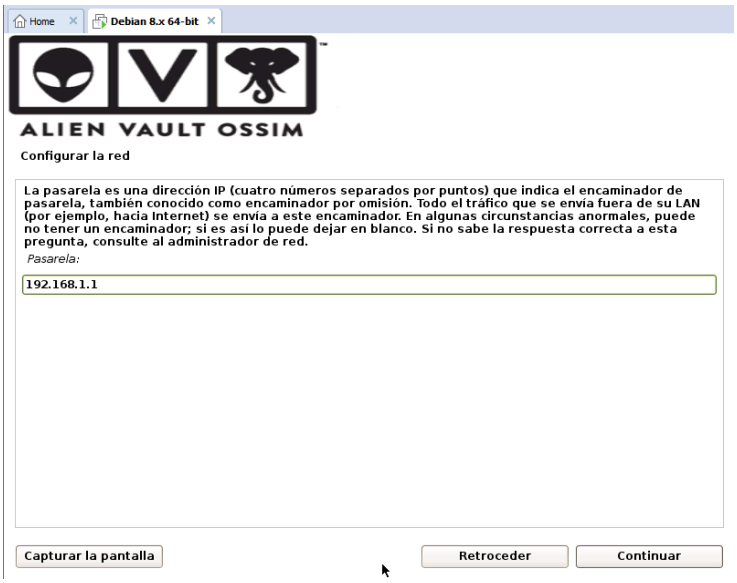


Figura B-8: Encaminador de Red

Configuración de los servidores de nombres si procede:

Figura B-9: Servidores de Nombres

Configuración de la red seleccionada:

Figura B-10: Configuración de Red

Configuración del usuario root:

Figura B-11: Usuario Administrador

Selección de la zona horaria:

Figura B-12: Zona Horaria

Instalación del sistema base:

Figura B-13: Instalación del Sistema

C Integración fuente en ArcSight

ArcSight es una de las tecnologías más potentes actualmente a nivel de SIEM. Se muestra cómo integrar una tecnología para así recibir sus logs y correlarlos.

Para integrar una tecnología en ArcSight se debe tener una máquina virtual además de la máquina de ArcSight. En dicha máquina virtual se recibirán los eventos de las tecnologías a integrar y se instalarán los conectores que serán los encargados de parsear los eventos y mandárselos al ArcSight.

Paso 1

Mandar los eventos de la tecnología a la máquina virtual que va a tener los conectores de ArcSight. Los eventos se pueden mandar de varias formas, peticiones POST hacia la máquina que se recogen mediante un php, mediante el almacenamiento de logs en un fichero, via syslog…etc.

El ejemplo que se va a explicar se configurará vía syslog. Se pueden tener múltiples tecnologías mandando los logs vía syslog, sin embargo, no se pueden tener varios conectores leyendo por el mismo puesto syslog. Para solventarlo, se puede mandar vía syslog pero cada tecnología a diferentes puertos.

Paso 2

Ejecución del .bin que contiene todos los conectores disponibles por ArcSight. Instalación del conector, pago genérico para todo tipo de conectores:

```
[root@carr Software]# ./ArcSight-7.7.0.8036.0-Connector-Linux64.bin
Preparing to install
Extracting the JRE from the installer archive...
Unpacking the JRE...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...

Launching installer...

Graphical installers are not supported by the VM. The console mode will be used instead...

===============================================================================
ArcSight SmartConnector                  (created with InstallAnywhere)
-------------------------------------------------------------------------------

Preparing CONSOLE Mode Installation...

===============================================================================
Introduction
------------
```

The ArcSight Installer will guide you through the installation of the ArcSight SmartConnector. The first step installs the core ArcSight SmartConnector components; then you select the ArcSight SmartConnector you wish to configure.

ArcSight recommends that you quit all other programs before continuing with this installation.

Respond to each prompt to proceed to the next step in the installation. If you want to change something on a previous step, type 'back'. To cancel this installation at any time, type 'quit'.

PRESS <ENTER> TO CONTINUE:

===
Choose Install Folder

Choose the folder where you would like to install an ArcSight SmartConnector. It is strongly recommended that you choose the folder name according to the device that you want to connect to, for example /ciscoids or /checkpointng. If you are upgrading an ArcSight SmartConnector from a previous version, please select the folder where the ArcSight SmartConnector is currently installed.

Where would you like to install?

Default Install Folder: /root/ArcSightSmartConnectors

ENTER AN ABSOLUTE PATH, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
: /opt/arcsight/conector_psyslog

INSTALL FOLDER IS: /opt/arcsight/conector_psyslog
IS THIS CORRECT? (Y/N): y

===
Choose Link Location

Where would you like to create links?

->1- Default: /root
2- In your home folder
3- Choose another location...

4- Don't create links

ENTER THE NUMBER OF AN OPTION ABOVE, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
: 4

===

```
Pre-Installation Summary
------------------------

Please Review the Following Information Before Continuing:

Product Name:
   ArcSight SmartConnector

Install Folder:
   /opt/arcsight/conector_psyslog

Link Folder:
   DO NOT INSTALL

PRESS <ENTER> TO CONTINUE:

===============================================================
Installing...
-------------

[==================|==================|==================|==========
==========]
 [------------

===============================================================
===============
Please Wait
-----------

===============================================================
===============
Please Wait
-----------
------|------------------|------------------|------------------]

===============================================================
==============
Please Wait
-----------

===============================================================
===============
Installation Complete
---------------------

The core components of the ArcSight SmartConnector have been successfully
installed to:

/opt/arcsight/conector_psyslog
```

To finish the configuration of the SmartAgent, please go to the folder:

/opt/arcsight/conector_psyslog/current/bin/

and execute the script:

./runagentsetup.sh

PRESS <ENTER> TO EXIT THE INSTALLER:
[root@carr Software]#

Una vez instalado el conector, se debe configurar para el tipo de evento, por lo que ejecutamos él runagentsetup.sh.

[root@carr Software]# cd /opt/arcsight/conector_psyslog/current/bin/
You have new mail in /var/spool/mail/root
[root@carr bin]# ./runagentsetup.sh

Assuming ARCSIGHT_HOME: /opt/arcsight/conector_psyslog/current
Assuming JAVA_HOME: /opt/arcsight/conector_psyslog/current/jre

ArcSight Agent Setup starting...

Connector Setup Wizard starting in mode [CONSOLE]
[Thu Jan 11 19:32:38 CET 2018] [INFO] Checking for a running instance of connector...
[Thu Jan 11 19:32:39 CET 2018] [INFO] Starting up connector...

Connector Setup

What would you like to do?

0- Add a Connector
1- Set Global Parameters

Please select an option: [Add a Connector] [0..1/cancel] :0

Select the connector to configure

Type:
0- Amazon Web Services CloudTrail
1- Apache HTTP Server Access Multiple Folder File
2- Apache HTTP Server Error File
3- Apache Tomcat File
4- ArcSight Asset Import File

5- ArcSight CEF Cisco FireSIGHT Syslog
6- ArcSight CEF Encrypted Syslog (UDP)
7- ArcSight CEF Folder Follower Scanner
8- ArcSight Common Event Format File
9- ArcSight Common Event Format Hadoop
10- ArcSight Common Event Format Multiple File
11- ArcSight Common Event Format REST
12- ArcSight FlexConnector CounterACT
13- ArcSight FlexConnector File
14- ArcSight FlexConnector ID-Based DB
15- ArcSight FlexConnector JSON Folder Follower
16- ArcSight FlexConnector Multiple DB
17- ArcSight FlexConnector Multiple Folder File
18- ArcSight FlexConnector Regex File
19- ArcSight FlexConnector Regex Folder File
(N)ext - ---------- Next page ----------
Please select an option [0..19]: n
Type:
(P)rev - ---------- Previous page ----------
20- ArcSight FlexConnector REST
21- ArcSight FlexConnector Scanner DB
22- ArcSight FlexConnector Scanner Text Reports
23- ArcSight FlexConnector Scanner XML Reports
24- ArcSight FlexConnector Time-Based DB
25- ArcSight FlexConnector XML File
26- ArcSight Logger Streaming Connector
27- Blue Coat Proxy SG Multiple Server File
28- Box
29- Bro IDS NG File
30- CA SiteMinder Single Sign-On File
31- CA Top Secret for z/OS File
32- Cisco IronPort Email Security Appliance File
33- Cisco IronPort Web Security Appliance File
34- Cisco Secure IPS SDEE
35- Dell ChangeAuditor DB
36- Dell InTrust DB
37- Extreme Networks Dragon Export Tool File
38- Extreme Networks Dragon IDS File
39- F-Secure Anti-Virus File
(N)ext - ---------- Next page ----------
Please select an option [20..39]: n
Type:
(P)rev - ---------- Previous page ----------
40- Gemalto SafeNet ProtectDB File
41- HPE IPC DB
42- HPE OpenVMS File
43- HPE Operations Manager i Web Services
44- HPE Operations Manager Incident Web Service
45- HPE-UX Audit File
46- IBM DB2 Multiple Instance UDB Audit File

47- IBM eServer iSeries Audit Journal File
48- IBM Lotus Domino Web Server File
49- IBM NVAS for z/OS File
50- IBM NVAS Session for z/OS File
51- IBM RACF for z/OS File
52- IBM SDSF for z/OS File
53- IBM SiteProtector DB
54- IBM System Log for z/OS File
55- IBM WebSphere File
56- IDMEF XML File
57- IP Flow (NetFlow/J-Flow)
58- IPFIX (IP Flow Information Export)
59- JBoss Security Audit File

(N)ext - ---------- Next page ----------
Please select an option [40..59]: n
Type:
(P)rev - ---------- Previous page ----------

60- Juniper Steel-Belted Radius File
61- Kaspersky DB
62- Linux Audit File
63- Lumension PatchLink Scanner DB
64- McAfee ePolicy Orchestrator DB
65- McAfee Network Security Manager DB (ID-based)
66- McAfee Network Security Manager DB (Time-based)
67- McAfee Vulnerability Manager DB
68- McAfee Web Gateway File
69- Microsoft Audit Collection System DB
70- Microsoft DHCP File
71- Microsoft DNS Trace Log Multiple Server File
72- Microsoft Exchange Message Tracking Log Multiple Server File
73- Microsoft Forefront DB
74- Microsoft Forefront Protection Server Management Console DB
75- Microsoft Forefront Threat Management Gateway File
76- Microsoft IIS Multiple Server File
77- Microsoft Network Policy Server File
78- Microsoft Office 365
79- Microsoft SharePoint Server DB

(N)ext - ---------- Next page ----------
Please select an option [60..79]: n
Type:
(P)rev - ---------- Previous page ----------

80- Microsoft SQL Server Multiple Instance Audit DB
81- Microsoft System Center Configuration Manager DB
82- Microsoft System Center Operations Manager DB
83- Microsoft Windows Event Log - Unified
84- NetApp Filer Event Log
85- NetIQ Security Manager DB
86- Nmap XML File
87- Novell Nsure Audit DB
88- Oracle Audit DB

89- Oracle Audit Vault DB
90- Oracle Audit XML File
91- Oracle SYSDBA Audit Multiple Folder
92- Oracle Unified Audit Trail DB
93- Oracle WebLogic Server File
94- OVAL XML File
95- PureSight Content-Filter DB
96- Qualys QualysGuard File
97- Rapid7 NeXpose XML File
98- Raw Syslog Daemon
99- SAINT Vulnerability Scanner File

(N)ext - ---------- Next page ----------
Please select an option [80..99]: n
Type:
(P)rev - ---------- Previous page ----------

100- SAP Real-Time Security Audit Multiple Folder File
101- SAP Security Audit File
102- sFlow
103- SNMP Unified
104- Snort Multiple File
105- Solsoft Policy Server
106- Sophos Anti-Virus DB
107- Sourcefire Defense Center eStreamer
108- Squid Web Proxy Server File
109- Sun ONE Directory Multiple Server File
110- Sun ONE Directory Server File
111- Sun ONE Web Access Multiple Server file
112- Sybase Adaptive Server Enterprise DB
113- Symantec AntiVirus Corporate Edition File
114- Symantec AntiVirus Corporate Edition Multiple File
115- Symantec Data Center Security DB
116- Symantec Endpoint Protection DB
117- Syslog Daemon
118- Syslog File
119- Syslog NG Daemon

(N)ext - ---------- Next page ----------
Please select an option [100..119]: n
Type:
(P)rev - ---------- Previous page ----------

120- Syslog Pipe
121- TCPDump
122- Tenable Nessus .nessus File
123- Tenable SecurityCenter XML File
124- Test Alert
125- TippingPoint SMS Syslog Extended
126- Trend Micro Control Manager DB (Legacy)
127- Trend Micro Control Manager Multiple DB
128- Tripwire IP360 File
129- Tripwire Manager File
130- UNIX Login/Logout

```
      131-    VMware Web Services
Please select an option [120..131]: 117


Please verify the following parameters

Type: Syslog Daemon


Are the values correct [yes/no/back/cancel]?yes

------------------------------------------------------------------------------
Enter the parameter details


Network Port[514]: 4514
IP Address[(ALL)]: ALL
Protocol:
      0-    UDP
      1-    Raw TCP
Please select an option [0..1][UDP]: UDP
Forwarder:
      0-    true
      1-    false
Please select an option [0..1][false]: false


Please verify the following parameters

Network Port: 4514
IP Address: (ALL)
Protocol: UDP
Forwarder: false


Are the values correct [yes/no/back/cancel]?yes


  |                                    | 0%Verifying the parameters
  |########################################| 100%
------------------------------------------------------------------------------
Enter the type of destination


0-    ArcSight Manager (encrypted)
1-    ArcSight Logger SmartMessage (encrypted)
2-    ArcSight Logger SmartMessage Pool (encrypted)
3-    CEF File
4-    Event Broker
5-    CEF Syslog
```

6- CEF Encrypted Syslog (UDP)
7- CSV File
8- Raw Syslog

Please select an option: [ArcSight Manager (encrypted)] [0..8/back/cancel] :0

Enter the destination parameters

****
WARNING: Some of the required parameters will contain security
sensitive information. Do you want to hide the input for these
parameters from the screen?[yes/no]
(typically you would answer 'NO' only if you are using a slow
link (like a serial RS232 or a very slow network link) since
this may add additional delays to the connection. If you are
not sure, then select 'YES' or hit enter.
****
[yes]?yes
Input for private parameters will be hidden.

Manager Hostname: Nombre_Maquina_Arcsight
Manager Port[8443]:8443
User: admin
Password:

AUP Master Destination:
0- true
1- false
Please select an option [0..1][false]: false
Filter Out All Events:
0- true
1- false
Please select an option [0..1][false]: false
Enable Demo CA:
0- true
1- false
Please select an option [0..1][false]: false

Please verify the following parameters

Manager Hostname: Nombre_Maquina_Arcsight
Manager Port: 8443
User: admin
Password: ********
AUP Master Destination: false
Filter Out All Events: false

Enable Demo CA: false

Are the values correct [yes/no/back/cancel]?yes

Enter the connector details

Name[]: psyslog
Location[]:
DeviceLocation[]:
Comment[]:

Please verify the following parameters

Name: psyslog
Location:
DeviceLocation:
Comment:

Are the values correct [yes/no/back/cancel]?yes

Registering destination
|##| 100%

Following certificate will be imported into connector trust store:
Host/port: Nombre_Maquina_Arcsight_8443
Details: CN= Nombre_Maquina_Arcsight, OU=ESM, O=Arcsight, L=95014, ST=CA, C=US

0- Import the certificate to connector from destination
1- Do not import the certificate to connector from destination

Please select an option: [Import the certificate to connector from destination] [0..1/back/cancel] :0

| | 0%Importing certificate, registering destination and restarting the container
|##| 100%

Add connector Summary
Following are the added connector details:
Connector Name [psyslog], Connector Type [syslog]

Continue [yes] ?yes

The Smart Connector is currently installed as a standalone application

0- Install as a service
1- Leave as a standalone application

Please select an option: [Install as a service] [0..1/back/cancel] :0

Specify the service parameters

Service Internal Name[syslog]: psyslog
Service Display Name[Syslog Daemon]: psyslog
Start the service automatically:
0- Yes
1- No
Please select an option [0..1][Yes]:

Please verify the following parameters

Service Internal Name: psyslog
Service Display Name: psyslog
Start the service automatically: Yes

Are the values correct [yes/no/back/cancel]?yes

Install Service Summary

The ArcSight SmartConnector is now configured to run as a service.

You can now start the SmartConnector by:

Going to the services application and starting the service:

ArcSight psyslog

Continue [yes] ?yes

Would you like to continue or exit?

0- Continue
1- Exit

Please select an option: [Continue] [0..1/back/cancel] :1

[Thu Jan 11 19:41:51 CET 2018] [INFO] Shutting Down Agent Framework Version [7.7.0.8036.0]

Shutting down Agent Modules now...

Shutting down Agent Setup Wizard...done.
[root@carr bin]#

El conector se ha instalado correctamente via syslog por el puerto 4514 udp. Al tener el servicio levantado, levanta automática el puerto 4514 por el puerto udp

Paso 3

Insertamos un fichero con el parser de los eventos en el directorio:

conector_psyslog/current/user/agent/flexagent/

Ejemplo de parser:

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=.*?(\\w+.\\w+.\\w+)\\|(\\d*)\\|(\\d*)\\|(\\d*.\\d*)?\\|(\\d*.\\d*)?\\|:(\\d*.\\d*)?\\|(\\d*.\\d
*)\\|(\\d+.\\d+)?\\|(\\w+).*?
token.count=9

token[0].name=dom
token[0].type=String

token[1].name=date
token[1].type=Integer

token[2].name=hour
token[2].type=Integer

token[3].name=tam
token[3].type=Double

token[4].name=max
token[4].type=Double

token[5].name=min
token[5].type=Double

token[6].name=rate
token[6].type=Double

token[7].name=media
token[7].type=Double

token[8].name=country
token[8].type=String

event.flexNumber1Label = __stringConstant("hour")
event.flexNumber1= hour
```

event.flexNumber2Label = __stringConstant("date ")
event.flexNumber2= date

event.deviceCustomNumber1Label = __stringConstant("max")
event.deviceCustomNumber1=max

event.deviceCustomNumber2Label = __stringConstant("min")
event.deviceCustomNumber2=min

event.deviceCustomNumber3Label = __stringConstant("rate")
event.deviceCustomNumber3=rate

event.deviceCustomString6Label = __stringConstant("media")
event.deviceCustomString6= media

event.flexString1Label=__stringConstant("dom")
event.flexString1= dom

event.name=__stringConstant("Eventos")

Paso 3

Levantamos el conector

/etc/init.d/psyslog start

Paso 4

Comprobar que se reciben los eventos parseados correctamente en el ArcSight.

D Plugin Alien Vault

AlienVault es otras de las tecnologías más potentes actualmente a nivel de SIEM. Se muestra cómo integrar una tecnología para así recibir sus logs y correlarlos.

Para integrar una tecnología en Alien Vault no es necesario tener una máquina previa, los eventos se mandar directamente vía syslog al Alien Vault y posteriormente hay que indicarle como identificar estos nuevos eventos para saber identificarlos y parsearlos correctamente.

Paso 1

Mandar por syslog los logs de la tecnología a integrar al AlienVault.

Paso 2

Segmentar los logs recibidos por syslog, para que sepa cuales vienen de la nueva tecnología. La segmentación se podría hacer por la IP del dispositivo que los manda.

Paso 3

Configurar el rotado de los ficheros para que no se hagan de un tamaña excesivo. (/etc/logrotate.d)

- **Configuration**

```
alienvault:/etc/logrotate.d# cat 3com-adsl-11g
/var/log/3com-adsl-11g.log {
        weekly
        missingok
        rotate 7
        compress
        notifempty
}
```

- **Restart rsyslogd**

Reiniciar el proceso syslog tras los cambios realizados.

```
# /etc/init.d/rsyslogd restart
```

Paso 4

Ejecución del fichero de configuración para escribir el plugin.

```
alienvault:/etc/ossim/agent/plugins# cat 3com-adsl-11g.cfg
;; 3Com ADSL 11g
;; plugin_id: 9001
;; type: detector
;;

[DEFAULT]
plugin_id=9001

[config]
type=detector
enable=yes
source=log
location=/var/log/3com-adsl-11g.log
create_file=false
process=
start=no
stop=no
startup=
shutdown=

[3com-adsl-11g-login-success]
#Feb 12 20:37:09 192.168.1.51  3Com ADSL 11g[3]:192.168.1.29 login success
event_type=event
regexp="(?P<date>\w{3}\s+\d{1,2}\s\d\d:\d\d:\d\d)\s+(?P<sensor>\S+)\s+3Com\sADSL\s
11g\[\d{1,2}\]:+(?P<src>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s+login\ssuccess"
date={normalize_date($date)}
sensor={resolv($sensor)}
plugin_sid=1
src_ip={$src}
```

Paso 5

Se prueba la regex con el fin de comprobar que se parsean los eventos de la tecnología correctamente. Esto es necesario para poder leer los datos en el correlador y generar alertas, informes…etc.

Type 1

```
# python
>>> import re
>>> r = re.match(
..."(?P<date>\w{3}\s+\d{1,2}\s\d\d:\d\d:\d\d)\s+(?P<sensor>\S+)\s+3Com\sADSL\s11g\[\.
... d{1,2}\]:+User\sfrom\s(?P<src>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s+timed\sout",
... "Feb 12 22:39:16 192.168.1.51  3Com ADSL 11g[10]:User from 192.168.1.29 timed
…out"
... )
>>> r.group(0) # Check that the entire string is processed
'Feb 12 22:39:16 192.168.1.51  3Com ADSL 11g[10]:User from 192.168.1.29 timed out'
```

```
>>> r.group(1) # Check that date is successfully split
'Feb 12 22:39:16'
>>> r.group(3) # Check source IP
'192.168.1.29'
```

Note: https://docs.python.org/2/library/re.html

Type 2

```
# /usr/share/ossim/scripts/regexp.py \
/var/log/3com-adsl-11g.log \
"(?P<date>\w{3}\s+\d{1,2}\s\d\d:\d\d:\d\d)\s+(?P<sensor>\S+)\s+3Com\sADSL\s11g\[\d{
1,2}\]:+(?P<src>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s+login\sfail" \
/etc/ossim/agent/plugins/3com-adsl-11g.cfg
(?P<date>\w{3}\s+\d{1,2}\s\d\d:\d\d:\d\d)\s+(?P<sensor>\S+)\s+3Com\sADSL\s11g\[\d{1
,2}\]:+(?P<src>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s+login\sfail
[('Feb 11 22:48:25', '192.168.1.51', '192.168.1.29')]
[('Feb 11 22:48:28', '192.168.1.51', '192.168.1.29')]
[('Feb 11 22:48:35', '192.168.1.51', '192.168.1.29')]
[('Feb 11 22:48:38', '192.168.1.51', '192.168.1.29')]
[('Feb 11 22:48:43', '192.168.1.51', '192.168.1.29')]
[('Feb 11 22:48:46', '192.168.1.51', '192.168.1.29')]
[('Feb 11 22:48:50', '192.168.1.51', '192.168.1.29')]
[('Feb 11 22:48:55', '192.168.1.51', '192.168.1.29')]
[('Feb 11 22:49:04', '192.168.1.51', '192.168.1.29')]
[('Feb 12 12:10:15', '192.168.1.51', '192.168.1.38')]
[('Feb 12 12:14:51', '192.168.1.51', '192.168.1.38')]
[('Feb 12 12:14:54', '192.168.1.51', '192.168.1.38')]
[('Feb 12 23:37:32', '192.168.1.51', '192.168.1.29')]
[('Feb 12 23:37:34', '192.168.1.51', '192.168.1.29')]
[('Feb 12 23:37:38', '192.168.1.51', '192.168.1.29')]
[('Feb 13 18:07:11', '192.168.1.51', '192.168.1.29')]
[('Feb 13 18:07:14', '192.168.1.51', '192.168.1.29')]
[('Feb 13 18:07:21', '192.168.1.51', '192.168.1.29')]
[('Feb 13 19:32:38', '192.168.1.51', '192.168.1.29')]
[('Feb 13 19:32:40', '192.168.1.51', '192.168.1.29')]
[('Feb 13 19:32:43', '192.168.1.51', '192.168.1.29')]
Counted 31 lines.
Matched 21 lines.
```

Paso 6

Se dan los permisos necesarios al fichero plugin.

```
# cd /etc/ossim/agent/plugins/
# chown root:www-data 3com-adsl-11g.cfg
```

Paso 7

Declaración del plugin.

- **config.cfg**

Editamos el fichero /etc/ossim/agent/config.cfg e insertamos el plugin que se ha desarrollado en la sección de plugins:

```
...
[plugins]
...
sudo=/etc/ossim/agent/plugins/sudo.cfg
whois-monitor=/etc/ossim/agent/plugins/whois-monitor.cfg
wmi-monitor=/etc/ossim/agent/plugins/wmi-monitor.cfg
3com-adsl-11g=/etc/ossim/agent/plugins/3com-adsl-11g.cfg
...
```

- **Database**

Se crea el script SQL para cargar el plugin en la base de datos.

```
# cat /usr/share/doc/ossim-mysql/contrib/plugins/3com-adsl-11g.sql
-- 3Com ADSL 11g
-- Plugin id:9001

DELETE FROM plugin WHERE id = "9001";
DELETE FROM plugin_sid where plugin_id = "9001";

INSERT IGNORE INTO plugin (id, type, name, description) VALUES (9001, 1, '3Com ADSL 11g', '3Com ADSL 11g');
INSERT IGNORE INTO plugin_sid (plugin_id, sid, category_id, class_id, name, priority, reliability) VALUES (9001, 1, NULL, NULL, '3Com-ADSL-11g: login success', 1, 3);
INSERT IGNORE INTO plugin_sid (plugin_id, sid, category_id, class_id, name, priority, reliability) VALUES (9001, 2, NULL, NULL, '3Com-ADSL-11g: login fail', 1, 3);
INSERT IGNORE INTO plugin_sid (plugin_id, sid, category_id, class_id, name, priority, reliability) VALUES (9001, 3, NULL, NULL, '3Com-ADSL-11g: forced logout', 1, 3);
INSERT IGNORE INTO plugin_sid (plugin_id, sid, category_id, class_id, name, priority, reliability) VALUES (9001, 4, NULL, NULL, '3Com-ADSL-11g: timed out', 1, 3);
```

Se carga en la base de datos.

```
# ossim-db < /usr/share/doc/ossim-mysql/contrib/plugins/3com-adsl-11g.sql
```

Se activa el plugin en el la parte del servidor, reseteando el proceso server del correlador.

```
# /etc/init.d/ossim-server restart
```

Activar el plugin en el agente, reseteando el proceso agente del correlador.

/etc/init.d/ossim-agent restart

Comprobar que el plugin ha sido integrado satisfactoriamente tanto a nivel web como a nivel de base de datos.

Database:
ossim-db
mysql> select * from plugin where id = 90009;
+----------+-------+------+----------------------+--------------------+--------------+--------+
| ctx | id | type | name | description | product_type | vendor |
+----------+-------+------+----------------------+--------------------+--------------+--------+
| | 90009 | 1 | AlienVault HIDS-ossec | Ossec Agent Status | 0 | NULL |
+----------+-------+------+----------------------+--------------------+--------------+--------+
1 row in set (0.00 sec)

mysql> select * from plugin_sid where plugin_id = 90009;
+----------+---------+---------+---------+---------+--------+----------+--------+----------+---------+
|plugin_ctx|plugin_id|sid|class_id|reliability|priority|name|aro|subcategory_id|category_id |
+----------+---------+---------+---------+---------+--------+----------+--------+----------+---------+
	90009	1	NULL	1	3	AlienVault HIDS:"Agent Status Active"	0.0000	NULL	NULL
	90009	2	NULL	1	1	AlienVault HIDS:"Agent Status Disconected"	0.0000	NULL	NULL
	90009	20000000	NULL	2	2	Alien VaultHIDS-ossec: Generic event	0.0000	NULL	NULL
	90009	2000000000	NULL	2	2	AlienVaultHIDS-ossecGeneric event	0.0000	NULL	NULL
+----------+---------+---------+---------+---------+--------+----------+--------+----------+---------+
4 rows in set (0.01 sec)

Web:

Configuration > THREAT INTELLIGENCE > DATA SOURCES

- **Comprobar que hay eventos en el SIEM**

ANALYSIS > SECURITY EVENTS (SIEM)

Paso 8

Se debe escribir una directiva. Editar el fichero user.xml que acaba de ser creado y añadir:

alienvault:/etc/ossim/server/7221e585-3a54-11e2-bfba-00145e16125b# cat user.xml
<?xml version="1.0" encoding="UTF-8"?>
...
<directive id="500001" name="AV Bruteforce attack, 3Com ADSL 11g" priority="4">
<rule type="detector" name="3Com ADSL 11g login failed" from="ANY" to="ANY" port_from="ANY" port_to="ANY" reliability="2" occurrence="1" plugin_id="9001" plugin_sid="2">

```
            <rules>
                <rule type="detector" name="3Com ADSL 11g login failed" from="1:SRC_IP" to="1:DST_IP" port_from="ANY" port_to="ANY" reliability="6" occurrence="3" time_out="120" plugin_id="9001" plugin_sid="2">
                    <rules>
                        <rule type="detector" name="3Com ADSL 11g login failed" from="1:SRC_IP" to="1:DST_IP" port_from="ANY" port_to="ANY" reliability="8" occurrence="10" time_out="360" plugin_id="9001" plugin_sid="2">
                            <rules>
                            <rule type="detector" name="3Com ADSL 11g login failed" from="1:SRC_IP" to="1:DST_IP" port_from="ANY" port_to="ANY" reliability="10" occurrence="1000" time_out="3600" plugin_id="9001" plugin_sid="2"/>
                            </rules>
                        </rule>
                    </rules>
                </rule>
            </rules>
        </rule>
</directive>
```

Resetear el correlador para aplicar los cambios realizados.

```
# /etc/init.d/ossim-server restart
```

Comprobar las directivas.

Configuration > THREAT INTELLIGENCE > DIRECTIVES

Comprobar las alertas del SIEM.

E Documento: Computer Security Threat Monitoring and Surveillance

El documento escrito por James P. Anderson en 1980 llamado Computer Security Threat Monitoring and Surveillance, forma parte de la historia de la informática, poniendo las bases para lo que posteriormente serán los fundamentos de las Seguridad Informática.

Se ha copiado el documento en el Anexo, algunas imágenes no se ven con nitidez al ser un documento fotocopiado.

James P. Anderson Co.
Box 42 Fort Washington, Pa. 19034

215 646-4706

COMPUTER SECURITY THREAT

MONITORING AND SURVEILLANCE

CONTRACT 79F296400

February 26, 1980

Revised:

April 15, 1980

Computer Security Threat Monitoring and Surveillance

February 26, 1980 - Revised: April 15, 1980

l.1 Introduction

This is the "final report of a study, the purpose of which was to improve the computer security auditing and surveillance capability of the customer's systems.

l.2 Background

Audit trails are taken by the customer on a relatively long term (weekly or monthly) basis. This data is accumulated in conjunction with normal systems accounting programs. The audit data is derived from SMF records collected daily from all machines in the main and Special Center. The data is temporarily consolidated into a single file ("dump" data set) from which the various summary accounting and audit trail reports are produced. After the various reports are generated, the entire daily collection of data is transferred to tape. several years of raw accounting data from all systems are kept in this medium.

Audit trail data is distributed to a variety of individuals for review: a DAC for GIMS applications, activity security officers for some applications located under their purview, but the majority to the customers data processing personnel! For the most part the users and sponsors of a data base or an application are not the recipients of security audit trail data.

Security audit trails can play an important role in the secU4ity program for a computer system. As they are presently structured, they are useful primarily in detecting unauthorized access to files. The currently collected customer audit trails are designed to detect unauthorized access to a dataset by user identifiers. However, it is evident that such audit trails are not complete Users (particularly ODP "personnel" with direct programming access to datasets) may operate at a level of control that bypasses the application level auditing and access controls. In other systems, particularly data management systems, the normal mode of access is expected to be interactive. Programmers with the ability to use access method primitives can frequently access database files directly without leaving any trace in the application access control and audit logs. Under the circunstances, such audit trail concepts can do little ~re than attempt
to detect frontal attacks on some system resource.

Security audit trails can play an important role in a security program for a computer system. As audit trails are presently structured on most machines, they are only useful primarily in detecting unauthorized access to files. For those computers which have no access control mechanisms built into the primary operating systems, the audit trail bears the burden of detecting unauthorized access to system resources. As access control mechanisms are installed in the operating systems, the need for security audit trail data will be even greater; it will not only be able to record attempted unauthorized access, but will be virtually the only method by which user actions which are authorized but excessive can be detected.

1.3 Summary

In computer installations in general, security audit trails, if taken, are rarely complete and almost never geared to the needs of the security officers whose responsibility it is to protect ADP assets. The balance of this report outlines the considerations and general design of a system which provides an initial set of tools to computer system security officers for use in their jobs. The discussion does not suggest the elimination of any existing security audit data collection and distribution. Rather it suggests augmenting any such schemes with information for the security personnel directly involved.

2. Threats

2.1 Scope

In order to design a security monitoring surveillance system, it is necessary to understand the types of threats and attacks that can be mounted against a computer system, and how these threats may manifest' themselves in audit data. It is also important to understand the threats and their sources from the viewpoint of identifying other data. It is also important to understand the threats and their sources from the viewpoint of identifying other data sources by which the threat may be recognized.

To assist the reader, the following definitions are used in this paper:

Threat:

The potential possibility of a deliberate unauthorized attempt to:

a) access information

b) manipulate information

c) render a system unreliable or unusable

Risk:

Accidental and unpredictable exposure of information, or violation of operations integrity due to malfunction of hardware or incomplete or incorrect software design.

Vulnerability:

A known or suspected flow in the hardware or software design or operation of a system that exposes the system to penetration of its information to accidental disclosure.

Attack:

A specific formulation or execution of a plan to carry out a threat.

Penetration:

A successful attack~ the ability to obtain unauthorized (undetected} access to files and programs or the control state of a computer system.

In considering the threat problem, the principal breakdown of threats is on the basis of whether or not an attacker is normally authorized to use the computer system, and whether or not a user of the computer system is authorized to use a particular resource in the system. The cases of interest are shown in Figure 1.
Another view of the representation of threats is shown in Figure 2. This representation shows the protected resources. surrounded by· rings of control and rings of "users" o In some ways this representation is more useful for purposes of identifying where and what kind of audit data might be of use in detecting the exercise of one of the threats shown.

2. 2 Gaining Access ·to the System - External Penetration

In the context of this report, the term "external penetration" is not confined to the usual case of an outsider attempting to gain access to a computer resource in an organization of which he is not a part. The term is meant to convey, in addition to the previous case, the notion of an employee of the organization who has physical access to the building housing the computer system but who is not an authorized computer user. These cases are of general and specific interest in that they represent in some ways the extremes of the problem of gaining access to a computer.

The true outsider has the most difficult task in some ways if the only means (terminals, RJE stations, etc.) of accessing a computer are physically co-located with the computer in the same buildings. Where access to computer resources is granted through wire communications, the external penetrator has a substantially easier task in attempting to

gain physical access. For those systems and networks has merely to wire tap a communication line to effectively gain use of the targeted system.

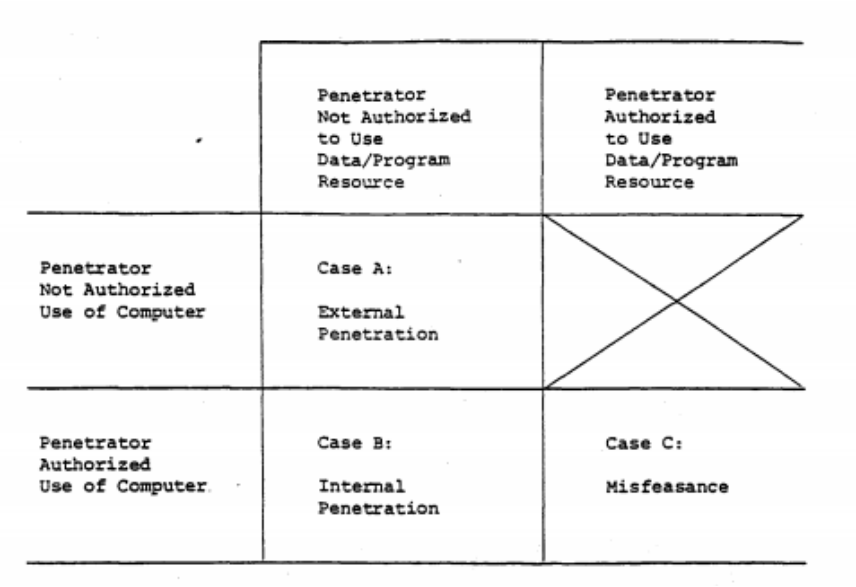


Figura 1: General Cases of Threats

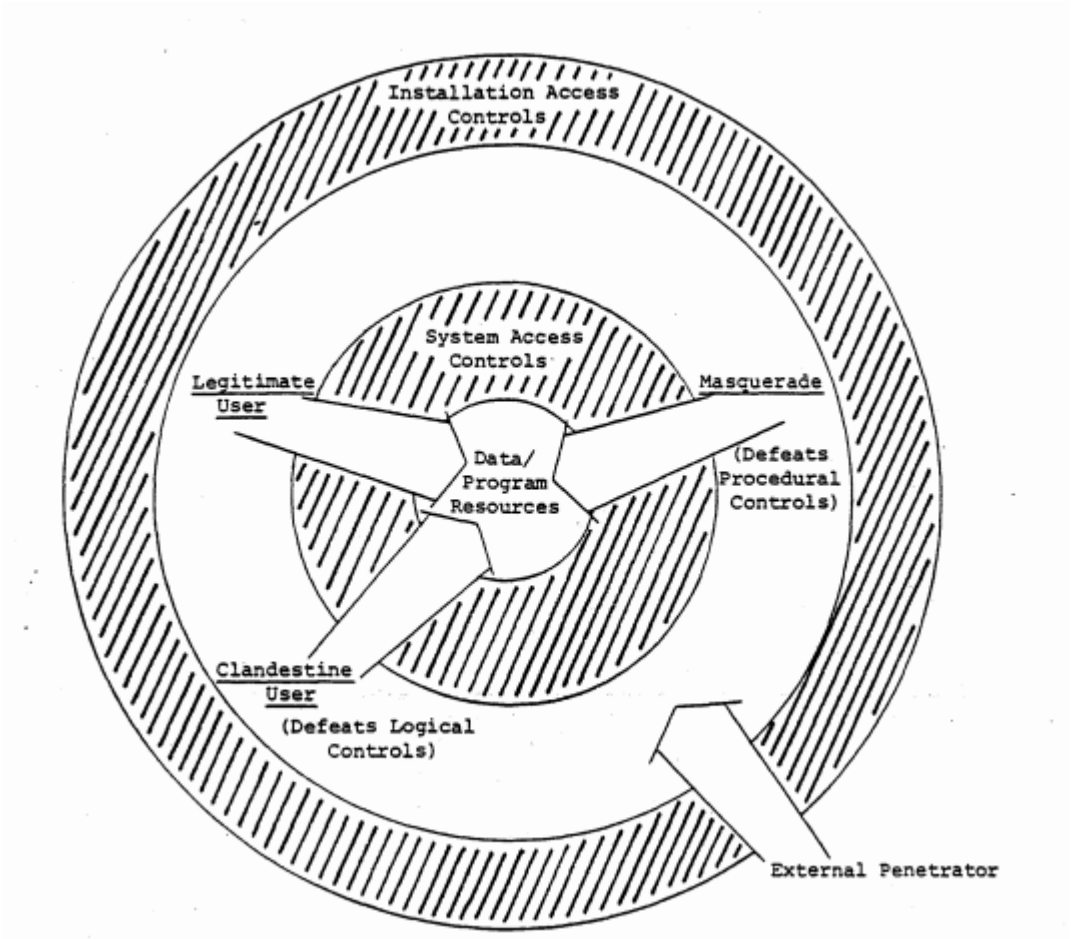


Figura 2: Threat Representations

The individual with physical access to the building housing the computer systems or its terminals does not have to ·resort to such exotic methods. However, it may be more difficult for such an individual to gain access to use the system without attracting attention. Whether or not this is true in any specific instance is in part a function of how mature the insolation is and in particular, whether or not there are many terminals for use of the computer resources.

In the case of the user with physical access to the building housing the computer systems, there is a possibility of additional information that may be useful to correlate for security purposes. As an example, in those buildings that employ security logging or building access systems that record the time and point of entry and exit of all individuals, it would be possible for detected security incidents to be correlated with individuals who could conceivably be involved in the incidents.

In case of unprotected communication lines, there is opportunity for individuals to attempt to gain use of computer systems by trail and error attempts at logging on. Records of the log on attempts if collected, would provide security officers with a substantial warning of unauthorized activity, and identification of at least the location from which unauthorized access is being attempted.

In most systems such data is not collected. This is because the systems are generally large with a large number of users, and recording the presumed attempted logons would consume too many system resources to warrant their acquisition.

In addition there is a potential problem created by recording in the audit data unsuccessful logons if those logons contain the password or other user authenticator. The danger is that the audit trail will contain partial or complete user authenticators or passwords from legitimate errors made by authorized users as well as the unsuccessful external penetration attempts. This is not to say such. data should not be collected, it is only to point out that in the collection it is possible that a greater danger is created.

Auditing of attempted logons can include identification of the terminal, the port through which the terminal is connected to the system, and the claimed identity of the user and the like. If the assets required it, it would be possible to trigger an immediate exception report to the security officer or other operations personnel if the number of unsuccessful longons from a given port number exceeded some threshold over time. The cost of this idea is the additional complication of maintaining logon records or even extracts from logon records on a per-port basis when the number of ports or the number of potential users of the system is extremely large. Note that the external penetrator threat translates into an internal threat as soon as the installation access controls have been penetrated.

2.3 Internal Penetration

In many installations, the internal penetration is more frequent than external penetrations. This is true for a variety of reasons, not the least of which is the internal penetrator has overcome a major barrier to unauthorized access, that is, the ability to gain use of a machine. Again for the purpose of identifying possible means of detection through audit trails, three classes of users can be identified. These are:

a. The masquerader
b. The legitimate user
c. The clandestine user

The user classes are shown in an order of increasing difficulty in detecting their activity through audit trail data. The ability, to detect activity of each category of user from audit data varies, in some cases considerably7 hence the breakdown.

2.3.1 The Masquerader

As indicated in the diagram, the masquerader is an internal user by definition. He can be any category of individual; either an external penetrator who has succeeded in penetrating the installation access controls, or an employee without full access to a computer system, or possibly an employee with full access to a computer system who wishes to exploit another legitimate users identification and password that he may have obtained.

This case is interesting because there is no particular feature to distinguish the masquerader from the legitimate user. Indeed, with possession of the proper user identifier and password, he is a legitimate user as far as the computer system is concerned. Masquerade is interesting in that it is by definition an "extra" use of a system by the unauthorized user. As such it should be possible to detect instances of such use by analysis of audit trail records to determine:

a. Use outside of normal time
b. Abnormal frequency of use
c. Abnormal volume of data reference
d. Abnormal patterns of reference to programs or data

As will be discussed in the subsequent section, the operative word is "abnormal" which implies that there is some notion of what "normal" is for a given user.

In attempting to detect masquerade, a surveillance system focuses on the legitimate user as the resource being "protected". In other types of surveillance the resource being protected may be other elements of the system such as devices, specific files and databases or programs and the like.

Quite obviously the masquerader can have as his intent any of the various stated purposes of penetration. Again, since his use of a system will be extra, that is in addition to normal use by a user of the same user number, this extra use can or should be detectable.

2.3.2 Legitimate User

The legitimate user as a threat to information resources is a case of misfeasance in that it involves the misuse of authorized access both to the system and to its data. Since the user is authorized to use the system, the audit trail records would not be expected to exhibit any abnormal patterns of reference, logon times and so forth. It is for this reason that the degree of difficulty in detecting "abnormal" use by a legitmate user of a system is more difficult than the preceding case. There maybe no "extra" use of resources that can be of help in detecting the activity.

It must be recognized that small amounts of misuse of authorized access would not be detected under any circumstance. As an instance, if the authorized user misuses his authority slighty, to print Snoopy calendars or to extract two extra records of data that he is otherwise authorized to use, a statistically satisfactory method of detecting such minor abnormalities is probably not feasible.

If the legitimate user makes use of his authorized access to refer to or gain access to information that is normally authorized in the conduct of his job, the audit trail should be able to reflect this. Similarly, if the authorized user misuses his access to gain large amounts of information by transferring many records or use an "excessive" amount of computer time, this too should be detectable. Initially, it may not be possible to detect a difference between a case of misfeasance and a masquerade. In general, it would be expected that the masquerade would show up as an anomaly in the time of use of a system whereas misfeasance would show up by one or more of the parameters total time used, or data transferred exceeding previously established norms.

2.3.3 Clandestine User

The clandestine user is quite possibly the most difficult to detect by normal audit trail methods. The assumption regarding clandestine users is that the user has or can seize supervisory control of the machine and as such can either operate below the level at which audit trail aata is taken or can use privileges or system primitives to evade audit trail data being recorded for him. As far as most audit trail information is concerned, the clandestine user is "the little man who isn't there". There is nothing that can be done to detect this type of user unless he activates his clandestine operations in a masquerade or as misfeasance of a legitmate user that may then create individual records that show up under those categories of use.

The clandestine user who effects a technical penetration to obtain control of the most privileged state the computer system, is not capable of being audited. Where the threat of such penetrations is considered high it would be possible to augment the internal auditing mechanisms of the individual computer with external measurements of busy or idle states of the CPU, the memory, secondary storage and so forth, and from this additional data possibly (avery weak possibly) detect "pure" phantom use.

2.3.4 Clandestine User Countermeasures

The penetration issue is one which can be played measure – countermeasure through what appears to be endless variations. What is really at the heart of the difficulty of "defense" is the fact that the penetrator has a myriad of places to effect operating system changes that permit penetration. At a high level of sophistication, the penetrator could temporarily alter the operating system to suppress audit recording of what he's doing. Depending on a number of factors, this is virtually impossible to detect purely by analysis of the internal audit records. It involves in looking for what isn't present. However, if ~e operating system changes for the penetration are only temporary, the changes could be detected, if the operating system code is continuously compared in some fashion with a reference version.

The security audit data is dependent to a large extent on the integrity of the origins of the audit trail records. The audit trails are a centralized recording of information originally

designed to support billing and other. accounting functions. To support security surveillance, the ideal situation would be to provide independent audit trails for each major component of the machine, preferably by a micro or other computer element associated with the device or devices supporting the use of the system.

Independent audit trails for each major component or function of a machine is derived from the experience of auditing in networks. It is clear that the suppression of audit records in a network where a number of points must be traversed through the network in order to affect the desired penetration, is virtually impossible unless one subverted every component of the network from the point of entry to the target and possibly back again. In sophisticated networks involving a transport layer, one or more systems as access systems and then server hosts, total control of all use recording of all such affected elements would not be possible. Under any circumstance, the distribution of recording among a number of points in a system greatly compounds the difficulty for the penetrator. In fairness, it must be pointed out that it also compounds the work for the compilers and users of audit trail data.

3. Characterization of Comouter Use

3.l Introduction

The basic premise of this study is that it is possible to characterize the use of a computer system by observing the various parameters available through audit trails, and to establish from these observations, "normal" ranges for the various values making up the characterizations.

3.2 The Unit of Computer Work - The Job or Session

Considering the problem of characterizing use of a computer the first issue that must be faced is what unit or units should be used to represent how a computer is used. It appears that the most natural unit of computer use is the notion of job in batch running or session in interactive working. Both of these terms denote a continuous unit or a single unit of use of a computer with a well-defined beginning and a well-defined end. The parameters that distinguish one unit from another are the user identifiers on whose behalf they are operated and the list of the program and (where available) data files entering into the program.

It should be noted that if the resource being monitored is the file or device that the notion of job or session as the principal parameter of characterization may not make much sense. In these instances, a list of references by user identifier or program (if such information is available) is the principal parameters of characterization of such use.

3.3 Time Parameters

There are basically 2 time parameters of interest that characterize how a system is used for a particular job. The first of these is the time of day (and in a larger sense the day of the week) that a particular job or session is operated. For many jobs this time of use is fixed within a fairly narrow range.

The second time parameter is the duration of length of time the job takes. While the fact that most modern systems are multi programmed and the elapsed real time for ~ job will vary accordingly, it is still a measure that one would ordinarily expect to have relatively little variability.

The time of day of the job initiation is one of the few use parameters with multiple values. Depending on the kind of user being characterized, the time of initiation of a particular task or job will vary, perhaps substantially. This is especially true in the case of interactive working where the choice of when to do a particular kind of task is totally up to the user under observation.

While system usage patterns can exhibit wide fluctuations from one user to another, it is expected that individual users establish patterns to their use of a system. It is these patterns that will be disturbed by masquerades.

Further, it should be evident that the ability to discriminate a particular indicator is a function of how daily the individuals own pattern of use fluctuates from day-to-day, and week-to-week.

This is well illustrated by the example given below where the ability to detect use of a resource outside of 'normal' time cannot be achieved if 'normal' time can be any hour of the day, any day of the week.

Detection of, outside of normal times of use is relatively straightforward. Individual jobs (sessions, job steps, etc.) are sorted on time of initiation and compared with previously recorded data for the specific user.

The basic question to be faced is the granularity of the analysis needed to detect 'out of time' use of a resource. For users exhibiting little variability in their use of a system, a gross measure, such as number of jobs (sessions, etc.), per quarter of the day (0000 - 0559, 0600 - 1159, ••• etc.) will be sufficient to discover second or third shift use of a system under the name of the subject under observation.

For another class of user, with considerable variability in time of use, it may be necessary to record usage by the hour. Obviously, if the 'normal' use is every hour of the day, the 'outside of normal time condition is not detectable. One would have to examine such users further to determine whether the normal use extends seven days a week, on holidays, through vacations, etc. Conceivably, 'normal' usage could extend through all of these periods. Then, the 'out of normal time' condition would not be a useful discriminant for that user.

Figure 2 shows the number of logons per hour for two different days (approximately 20 days apart) for a number of different users. Users I, II, and IV exhibit consistent patterns of logon, while users III and V exhibit more variability (in these two samples).

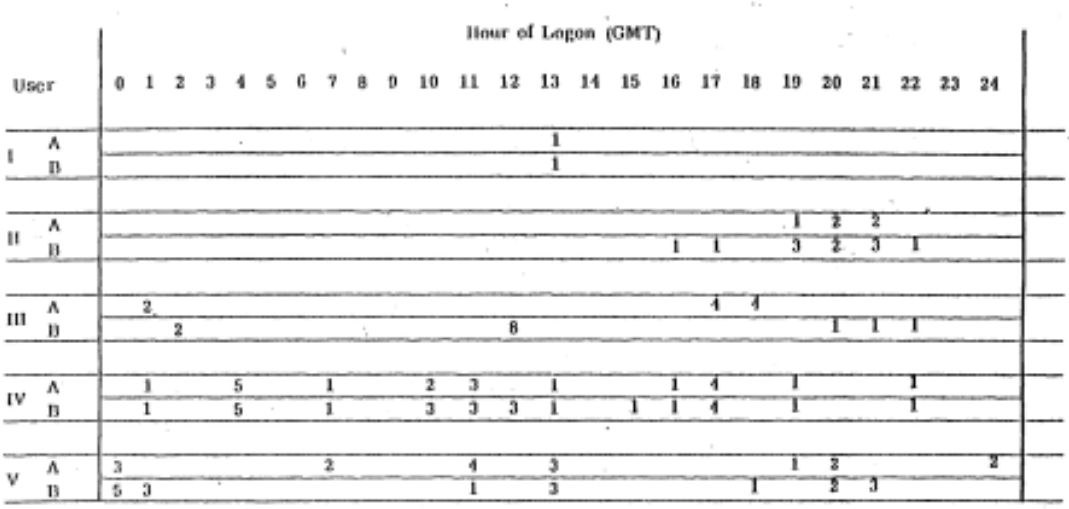


Figura 3: Number of Logons/Hr

If (for purposes of illustration) we assume that the 'A' data is the average (or cumulative) experience with the user in question, the variability in time of use could be scored by summing the squares of absolute values of the difference, i.e.,

$$score = \sum_{i=1}^{24} \left| (A_i - B_i) \right|^2$$

While not a particularly elegant measure, it does show for the several users represented, those whose logon pattern exhibit greatest variability, which might be the result of masquerade. Depending on other measures, those users might then become subjects of additional investigations.

The time of use abnormality scores for the five samples are:

User	Score
I	0
II	8
III	107
IV	11
V	41

Depending on where the cutoff point is set for reporting, one would expect to see 'III' and 'V' reported as being out of range.

In addition to the elapsed real time for a particular problem, we can measure the actual computer time used on a particular problem. This measure should not vary substantially, but a heavy system load which causes programs to be swapped in and out frequently can increase the elapsed running time for the problem. The increase should not be significant unless there is some other reason.

3.4 Dataset and Program Usage

The parameters that can be measured in this area varies significantly from one system to another. In some cases it is possible to identify the number of records read and written to a particular dataset or file while in another case on another system, the only data reference information that would be available would be a total number of pages transferred between a file system to a processor, with no indication being given whether those pages were read or written. These differences are a result of the fact that the audit data is taken for accounting purposes rather than security purposes, and as a consequence the kind of information that's collected is driven by accounting interests rather than what one would prefer for security purposes.

With regard to program usage the principal concern as far as security audit goes is whether or not a program was referred to for execution purposes or whether it is being read and written as data. This is significant for a security viewpoint because of the fact of reading and writing of programs as data is almost certainly a clue of penetration activity as opposed to normal system use. It must be understood that the reading and writing programs as data does not mean the results of compilation. Thus the principle data parameter for programs or data files is the number of records read or written.

3.5 Monitoring Files and Devices

The preceding discussion focused on the monitoring of a particular user identifier through the range of actions that the user identifier is allowed to do include submitting jobs, use of system and so forth. It is indeed the monitoring of system users that is the focus of the preceding kinds of surveillance and monitoring techniques. When one shifts the attention to monitoring a par~icular file or correspondingly a device, the kind of information collected, how it is collected and how it is used differs.

3.6 Group Statistics

While one could attempt to detect abnormal values of parameters against all of the job records for a single user, it is believed that better measures and better security can be obtained by grouping the job records into sets having the property that each job or session refers to the same set of files that is, an identical set of files.

The presumption is that the session or job referring to the same file sets can be considered to belong to the same population and will exhibit similar statistical properties from run to run. An arbitrary deviation of the norm for the user is a criterion for reporting a particular use and generating an "abnormal volume of data" or an "abnormal (measure of one of the parameters discussed above) exception". With no other data available, if the observed statistic for a parameter is more than plus or minus 2.58 standard deviations from the mean, it is out in the five percent range and probably is worthy of examination.

The abnormal patterns of reference are determined simply by discovery of file references that have not been previously encountered. If the files referenced in a particular job are not identical to a set previously seen, this should be reported as a new event. In the section on the organization of a surveillance system, some of these comments are, illustrated with the results of a model system.

4. Structure of a surveillance System

4.1 Introduction

This section outlines the functional components of a security monitoring and surveillance system. It identifies the key programs that will be required and considers a number of alternatives in implementing such a design. Figure 4 is a diagram of the central function of a surveillance system. It shows elements for the automatic generation of security exception reports.

4.1.1 Monitoring of Users

The diagram, Figure 4, shows the major steps involved in ·producing the monitoring ·and surveillance system data files. The first step is the selection of audit records affecting the element or elements being audited. This step is included in the overall design on the premise that the ability to keep history records for a large number of users will be storage limited. The second reason for including this is the premise that most use of a system is benign and proper and that for large populations, the bulk of the population is not of interest to the security personnel at any one time. In practice, a security office may have 50-100 "cases" in which they are interested. Some of these cases may be merely random selections from the total user population to be audited for a period of time, not with the intent of finding any wrong-doing, but with the intent of determining any possible wrong-doing.

4.1.2 Sorting Audit Records

The audit records selected in the previous step are then sorted on, a user identifier, and then within that, job identifier, date, time and so forth. The purpose of the sort is to collect together all records constituting a job. In most audit systems the job is represented by a number of audit records; job initiation, job termination, job execution, etc. The information of interest may be distributed over all the different kinds of records. The output of the sort is used as input to a program that builds session records.

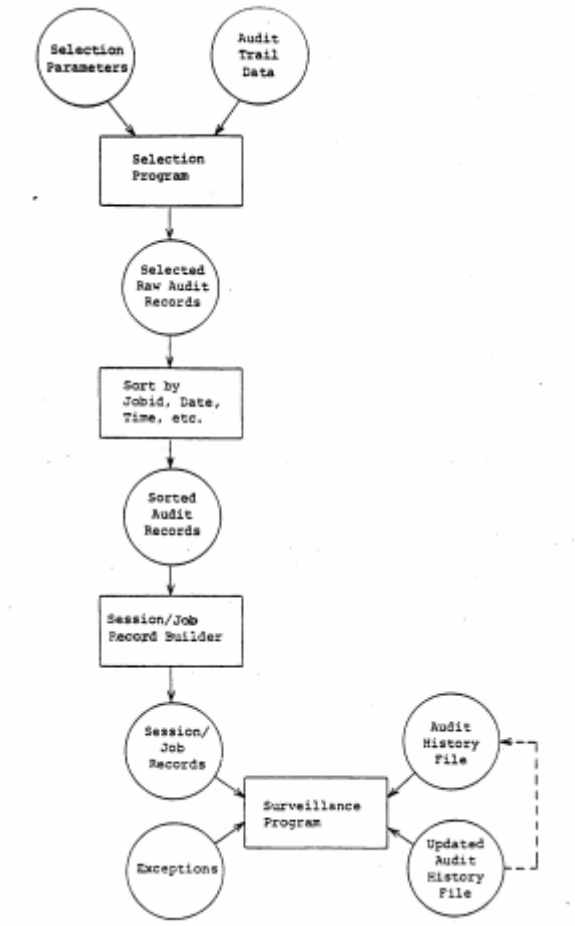


Figura 4: Surveillance system

4.1.3 Session Record Builder

Whether or not a session record builder is required, is a function of the type of audit data that is collected and possibly the type of system being employed. The model constructed as part of the project to determine the feasibility and the difficulty of doing surveillance of this type was based on a time sharing which provided a variety of records that required processing of all the records for a particular session in order to determine how much input and output had occurred. Other systems accumulate this information and make it available as part of a record identifying the termination of a job or program or as part of a program summary. The need for this step is a function of the underlying audit recording system for which it is built.

4.1.4 Surveillance Program

In some respects, this is the heart of the system in that to performs a variety of functions. In the prototype or model system, the surveillance system performs the following functions: It accumulates all instances of the same kind of job where job is defined in this case as having same program and file reference set involved (see 3.6). As it considers each job (or session) it compares the parameters measured on a session; that is the connect time, the number of input - output characters, the numbers of file references, etc., against a set of

absolute limits. The absolute limits were arbitrarily chosen by taking statistics over a large member of users and setting the limits such that it would cause an exception report if an individual session was unusual in and by itself.

In addition to the absolute limits, an individual session record is subject also to the distribution test. Distribution tests are those elements that are single values treated as samples, compared against distribution represented by the mean and the standard deviations of those means. If any of the parameters measured are greater than 2.58 standard deviations from the mean in either direction, the session record is reported as an exception. After these two operations are performed the session record is accumulated with all others like it and statistics for the set are available. Nothing is done with these statistics in prototype program. However, a similar measure could be employed to say how does the mean of all of the individual runs for this day compare with the accumulated mean, etc. inally the history master record is updated with the session summary data and the process repeated for the next set of session records.

In order to minimize file passes, the surveillance program recognizes when a master record has not been updated in fifteen days. This is an arbitrary time period established for the model program that is used to keep the history file at a reasonable size. In the event it finds such a record that has not been updated in fifteen days, it is removed from the history records, and reported as a record dropped for lack of activity.

Obviously with the records being dropped and added the other consideration is that a previous history record does not exist for a particular user. In this case, new master records are created and inserted in the correct place. No statistical reporting or distribution tests are performed in this case, but the absolute limits tests are recorded. In order to provide the security officer with some notion of what is going on, an exception report item is created for the session summary records that indicates that a new history master record is being created, and the new master record is available for display as part of the exception reporting.

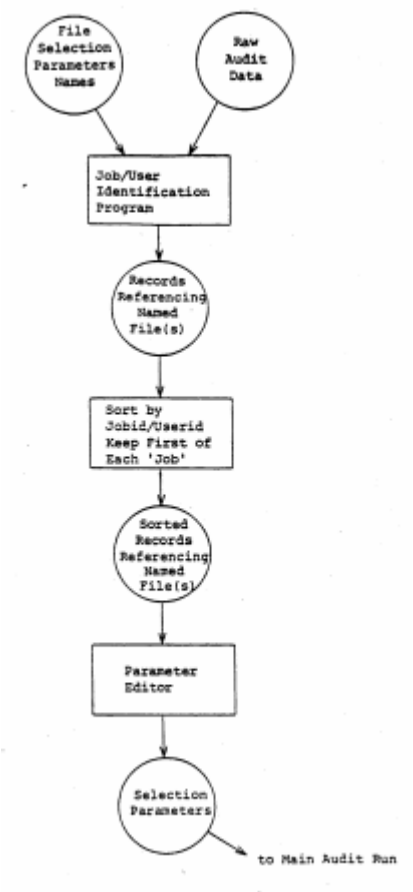


Figura 5: Processing to Audit File Use

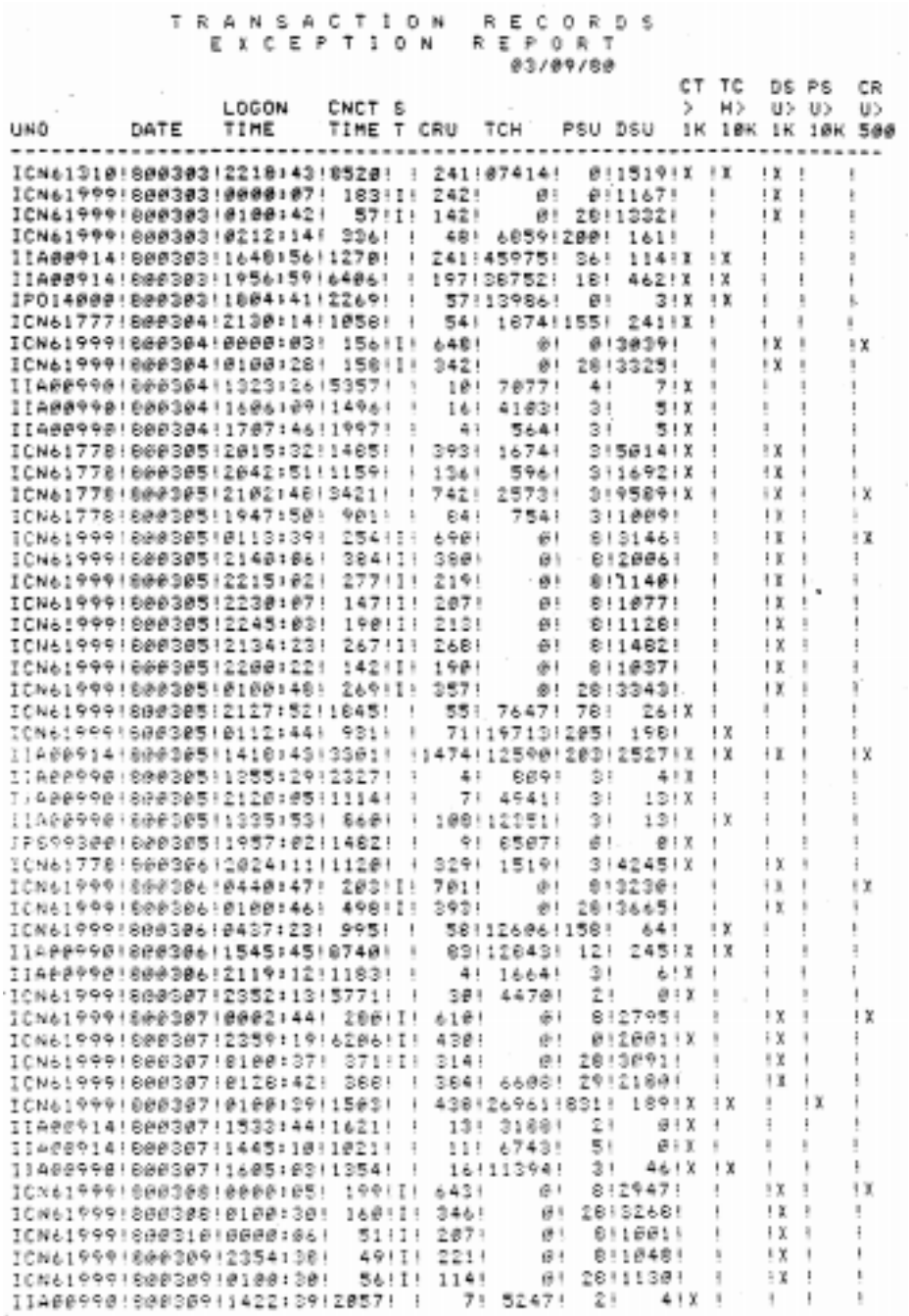
The entire sequence outlined above of selecting records of interest sorting them creating session summaries, updating master and the like and adding to the exception report is run once a day at the time the accounting files are turned over. The exception records are accumulated until such time as the reports are actually prepared. A sample of the reports from the model system are shown in figures 6,7,8 and 9.

4.2 Monitoring Files

Producing the records necessary to monitor use of files or other objects in a system is similar to that outlined above for monitoring users activities in a system. The principal difference is that fact that the element being sorted is the 'file', and the records being kept are on a per user basis. In some ways the files are a little more complicated than the users activities files in that multiple accesses to the same file in three or four different runs are to be treated in some sense differently, particularly in terms of the amount of data read from or written to the target file.

The file or device may require more than one pass of the audit file in order to collect the necessary information. As an example, if one wanted to record against a particular file, the users identifier and the session statistics associated with that reference to that file, it may be necessary to first pass the audit data file looking for those user identifiers or other session identifiers that are associated with its reference, make a list of those and then on a second pass of the audit data file collect the session records necessary to produce session summary statistics to be recorded against the file name. An example process flow is

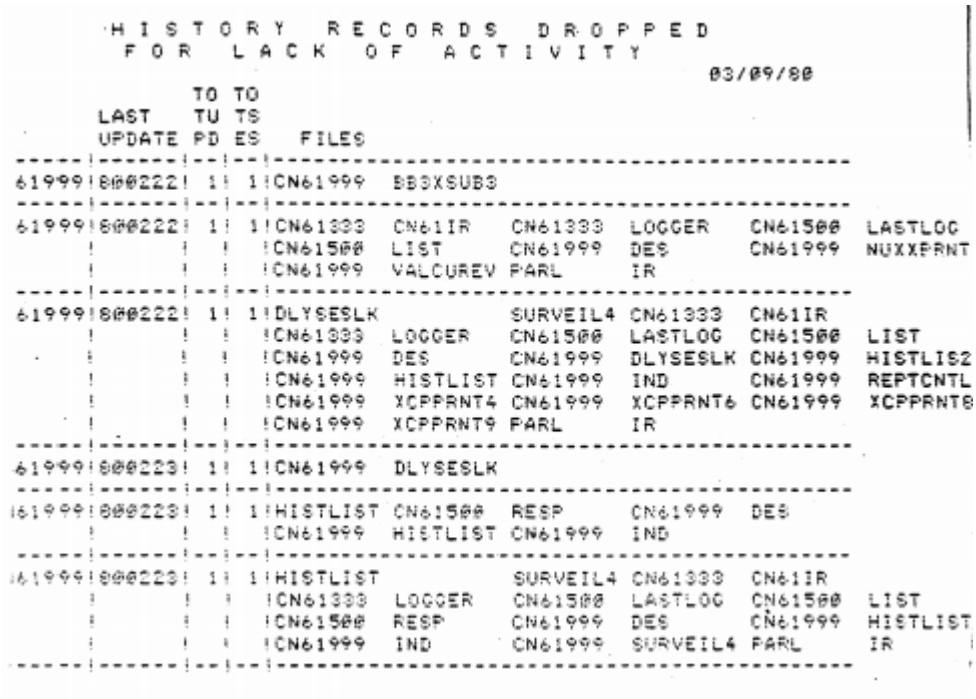
shown, Figure 5. Quite obviously these procedures vary as a function of the details of the type of audit trails being taken and the kind of monitoring that one attempts to perform on the specific objects.



TRANSACTION RECORDS
EXCEPTION REPORT
03/09/80

UNO	DATE	LOGON TIME	CNCT TIME	S T	CRU	TCH	PSU	DSU	CT > 1K	TC H> 10K	DS U> 1K	PS U> 10K	CR U> 500
ICN61310	800303	2210:43	0528	I	241	87414	0	1519	X	X	X		
ICN61999	800303	0000:07	183	I	242	0	0	1167			X		
ICN61999	800303	0100:42	57	I	142	0	20	1332			X		
ICN61999	800303	0212:14	336		48	6059	200	161					
IIA00914	800303	1640:56	1270		241	45975	36	114	X	X			
IIA00914	800303	1956:59	6406		197	38752	18	462	X	X			
IPO14000	800303	1004:41	2269		57	13986	0	3	X	X			
ICN61777	800304	2130:14	1058		54	1074	155	241	X				
ICN61999	800304	0000:03	156	I	648	0	0	3039			X		X
ICN61999	800304	0100:28	158	I	342	0	20	3325			X		
IIA00998	800304	1323:26	5357		10	7077	4	7	X				
IIA00998	800304	1606:09	1496		16	4103	3	5	X				
IIA00998	800304	1707:46	1997		4	564	3	5	X				
ICN61778	800305	2015:32	1405		393	16741	3	5014	X		X		
ICN61778	800305	2042:51	1159		136	596	3	1692	X		X		
ICN61778	800305	2102:48	3421		742	2573	3	9589	X		X		X
ICN61778	800305	1947:50	901		64	754	3	1009			X		
ICN61999	800305	0113:39	254	I	698	0	0	3146			X		X
ICN61999	800305	2140:06	384	I	388	0	0	2006			X		
ICN61999	800305	2215:02	277	I	219	0	0	1140			X		
ICN61999	800305	2230:07	147	I	207	0	0	1077			X		
ICN61999	800305	2245:03	190	I	213	0	0	1128			X		
ICN61999	800305	2134:23	267	I	268	0	0	1482			X		
ICN61999	800305	2200:22	142	I	198	0	0	1037			X		
ICN61999	800305	0100:48	269	I	357	0	20	3343			X		
ICN61999	800305	2127:52	1045		55	7647	78	26	X				
ICN61999	800305	0112:44	931		71	19713	285	198		X			
IIA00914	800305	1410:43	3301		1474	12590	283	2527	X	X	X		X
IIA00998	800305	1055:29	2327		4	889	3	4	X				
IIA00998	800305	2120:05	1114		7	4941	3	13	X				
IIA00998	800305	1335:53	660		100	12151	3	13		X			
IPS99300	800305	1957:02	1482		9	6507	6	8	X				
ICN61778	800306	2024:11	1120		329	1519	3	4245	X		X		
ICN61999	800306	0440:47	203	I	701	0	0	3230			X		X
ICN61999	800306	0100:46	498	I	393	0	20	3665			X		
ICN61999	800306	0437:23	995		58	12606	158	64		X			
IIA00998	800306	1545:45	0740		89	12043	12	245	X	X			
IIA00998	800306	2119:12	1183		4	1664	3	6	X				
ICN61999	800307	2352:13	5771		38	4470	2	0	X				
ICN61999	800307	0002:44	286	I	618	0	0	2795			X		X
ICN61999	800307	2359:19	6206	I	438	0	0	2001	X		X		
ICN61999	800307	0100:37	371	I	314	0	20	3091			X		
ICN61999	800307	0126:42	366		364	6606	29	2180			X		
ICN61999	800307	0100:39	1563		438	26961	831	189	X	X		X	
IIA00914	800307	1533:44	1621		13	3188	2	8	X				
IIA00914	800307	1445:18	1021		11	6743	5	8	X				
IIA00998	800307	1605:03	1354		16	11394	3	46	X	X			
ICN61999	800308	0000:05	199	I	643	0	0	2947			X		X
ICN61999	800308	0100:30	168	I	346	0	20	3268			X		
ICN61999	800310	0000:06	51	I	287	0	0	1001			X		
ICN61999	800309	2354:38	49	I	221	0	0	1048			X		
ICN61999	800309	0100:30	56	I	114	0	20	1130			X		
IIA00998	800309	1422:09	2857		7	5247	2	4	X				

Figura 6: Transaction Records Exception Report

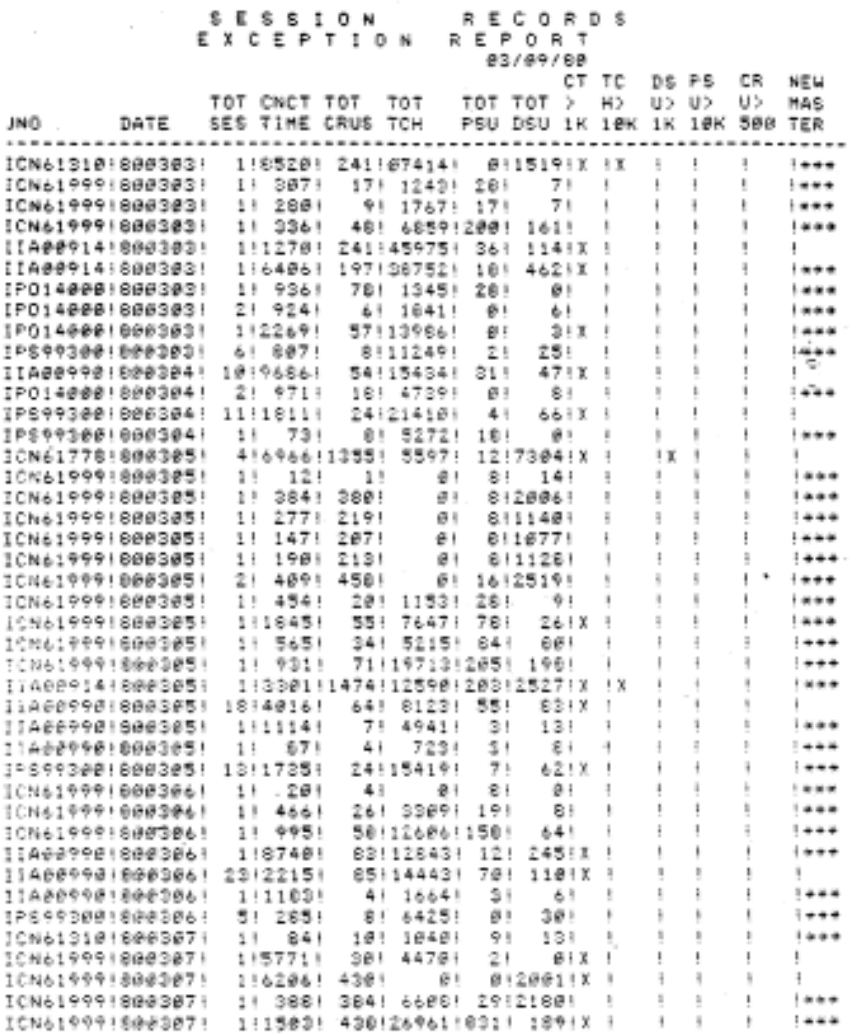


```
        HISTORY  RECORDS  DROPPED
          FOR  LACK  OF  ACTIVITY
                                                          03/09/80

             TO TO
      LAST   TU TS
      UPDATE PD ES   FILES
-----|------|--|--|-------------------------------------------
61999|800222| 1| 1|CN61999  BB3XSUB3
-----|------|--|--|-------------------------------------------
61999|800222| 1| 1|CN61333  CN61IR    CN61333  LOGGER   CN61500  LASTLOG
     |      |  |  |CN61500  LIST      CN61999  DES      CN61999  NUXXPRNT
     |      |  |  |CN61999  VALCUREV  PARL     IR
-----|------|--|--|-------------------------------------------
61999|800222| 1| 1|DLYSESLK           SURVEIL4 CN61333  CN61IR
     |      |  |  |CN61333  LOGGER    CN61500  LASTLOG  CN61500  LIST
     |      |  |  |CN61999  DES       CN61999  DLYSESLK CN61999  HISTLIS2
     |      |  |  |CN61999  HISTLIST  CN61999  IND      CN61999  REPTCNTL
     |      |  |  |CN61999  XCPPRNT4  CN61999  XCPPRNT6 CN61999  XCPPRNT8
     |      |  |  |CN61999  XCPPRNT9  PARL     IR
-----|------|--|--|-------------------------------------------
61999|800223| 1| 1|CN61999  DLYSESLK
-----|------|--|--|-------------------------------------------
61999|800223| 1| 1|HISTLIST CN61500  RESP     CN61999  DES
     |      |  |  |CN61999  HISTLIST CN61999  IND
-----|------|--|--|-------------------------------------------
61999|800223| 1| 1|HISTLIST           SURVEIL4 CN61333  CN61IR
     |      |  |  |CN61333  LOGGER    CN61500  LASTLOG  CN61500  LIST
     |      |  |  |CN61500  RESP      CN61999  DES      CN61999  HISTLIST
     |      |  |  |CN61999  IND       CN61999  SURVEIL4 PARL     IR
-----|------|--|--|-------------------------------------------
```

Figura 7: History Records Dropped For Lack of Activity

FIGURE 8



SESSION RECORDS
EXCEPTION REPORT
03/09/00

JNO	DATE	TOT SES	CNCT TIME	TOT CRUS	TOT TCH	TOT PSU	TOT DSU	CT > 1K	TC H> 10K	DS U> 1K	PS U> 10K	CR U> 500	NEW MAS TER
ICN61310	000303	1	9520	241	87414	0	1519	X	X				***
ICN61999	000303	1	307	17	1240	20	7						***
ICN61999	000303	1	288	9	1767	17	7						***
ICN61999	000303	1	336	48	6859	200	161						***
IIA00914	000303	1	1270	241	45975	36	114	X					
IIA00914	000303	1	6406	197	38752	10	462	X					***
IPO14000	000303	1	936	70	1345	20	0						***
IPO14000	000303	2	924	6	1041	0	6						***
IPO14000	000303	1	2269	57	13986	0	3	X					***
IPS99300	000303	6	887	8	11249	2	25						***
IIA00990	000304	10	9686	54	15434	31	47	X					
IPO14000	000304	2	971	18	4739	0	8						***
IPS99300	000304	11	1811	24	21410	4	66	X					
IPS99300	000304	1	73	0	5272	10	0						***
ICN61778	000305	4	6966	1355	5597	12	7304	X		X			
ICN61999	000305	1	12	1	0	8	14						***
ICN61999	000305	1	384	380	0	8	2006						***
ICN61999	000305	1	277	219	0	8	1140						***
ICN61999	000305	1	147	207	0	0	1077						***
ICN61999	000305	1	190	213	0	8	1126						***
ICN61999	000305	2	409	450	0	16	2519						***
ICN61999	000305	1	454	20	1153	26	9						***
ICN61999	000305	1	1845	55	7647	70	26	X					***
ICN61999	000305	1	565	34	5215	84	60						***
ICN61999	000305	1	931	71	19713	205	196						***
IIA00914	000305	1	3301	1474	12590	203	2527	X	X				***
IIA00990	000305	18	4016	64	8123	55	63	X					
IIA00990	000305	1	1114	7	4941	3	13						***
IIA00990	000305	1	67	4	723	3	8						***
IPS99300	000305	13	1735	24	15419	7	62	X					***
ICN61999	000306	1	201	4	0	8	0						***
ICN61999	000306	1	466	26	3309	19	8						***
ICN61999	000306	1	995	58	11606	150	64						***
IIA00990	000306	1	8740	83	12643	12	245	X					***
IIA00990	000306	23	2215	65	14443	70	110	X					
IIA00990	000306	1	1103	4	1664	3	6						***
IPS99300	000306	5	265	8	6425	0	30						***
ICN61310	000307	1	84	10	1040	9	13						***
ICN61999	000307	1	5771	30	4470	2	0	X					
ICN61999	000307	1	6206	430	0	0	2001	X					
ICN61999	000307	1	388	384	6606	29	2180						***
ICN61999	000307	1	1583	430	26961	803	189	X					***

Figura 8: Session Records Exception Report

Figura 9: New Master Records

5. Adapting to SMF Data

5.l Relevant SMF Records

The principal SMF records of use in performing the kind of auditing discussed in the preceding sections are record types 4, 5, 6, 10, 14, 15, 17, 18, 20, 25, 26, 34, 35, 40, 62, 63, 64, 67, 68, 69, SO and Sl. Ordinarily, these record types would be the records making up the details of a particular job or use of a computer. In producing the audit flow, selection parameters such as user names can be used to extract all audit trail data with that user name associated with it to provide input to the audit record sort step which collects together in one place all record types associated with a particular job or use of a computer. The output of sorted job records is used as input-to a job summary or session summary record builder. It is the summary record builder program that would provide the essential information from which the audit history records would be created and maintained.

When dealing with SMF, one is overwhelmed with data, a good deal of it not necessarily useful for security audit purposes. A basic audit history record is shown in Figure 10. This record is the one used in the model program. The individual data items are self-explanatory for the most part. The items indicated in square brackets are additional information available from SMF records that was not available in the accounting data in the model system.

Where the record shows sessions, one could substitute the notion of jobs; aside from that, the history records characterize a particular use of the computer system in which the model was being developed.

Data Item	Comments
USERID	
[JOBID]	
File/data set list	List of data sets referred to in this job (session).
[Number of read/writes to each data set]	
Total number of runs (sessions) to date	
Frequency count of logons (job run times) to date	Counted by quarter of day; other distributions are possible.
Date of last update	Used to determine when to purge audit history record.
Total number of updates	
Total to date of:	
. CPU time . I/O operations . Connect time (job turn-around time) . Characters transmitted to terminal	Used to compute mean values: = < parameter >/total sessions
Maximum/minimum to date of:	
. CPU time . I/O operations . Connect time . Characters transmitted	Establishes observed range of values.

NOTE: Items in square brackets ([]) were not available in model system.

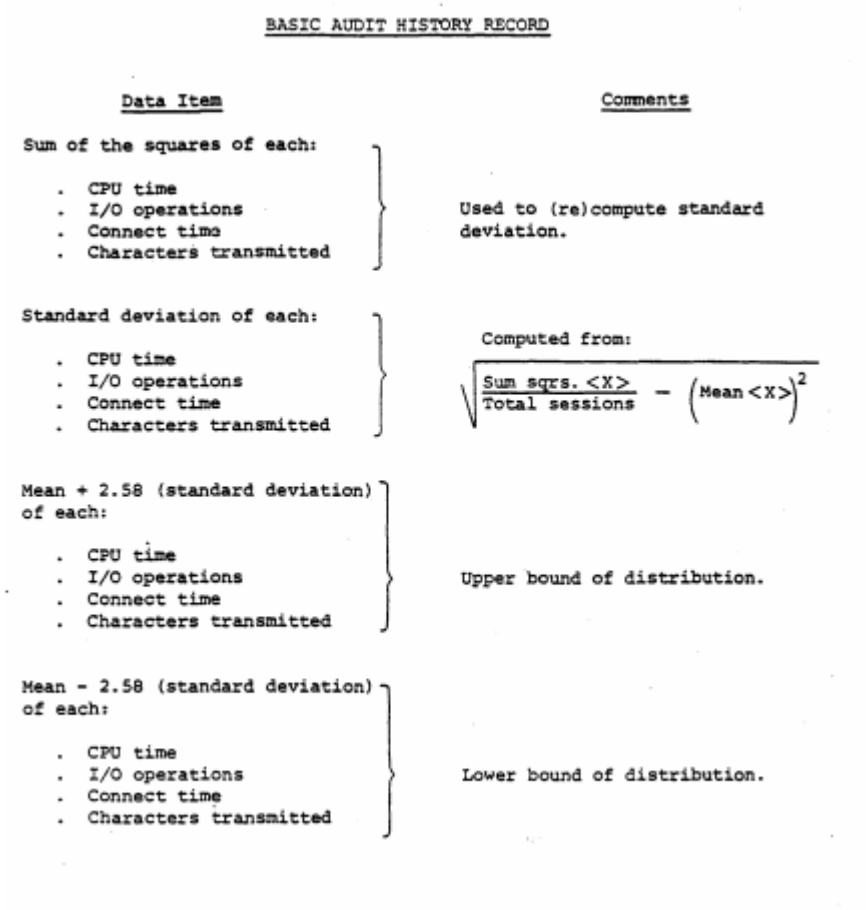


Figura 10: Basic History Record

Inclusion of the actual standard deviation values and the mean plus or minus 2.58 times the standard deviation of each of the major parameters was to simplify the computation and to make the program run a little faster. It is certainly feasible to compute this data each time it is required; however, with the large number of records, the computation time becomes excessive, and the value of storing it in the record itself becomes a little more apparent.

The accounting data available in the model system does not show the number of read and write operations to each data set that is referred to in the file data set list. If this data were available, the totals, the standard deviations, and the sum of squares information could be augmented by this data to provide a finer grain of detail in the audit history record. It would then be possible to make an exception report for and of those items that exceeded the bounds around the mean for each file rather than treating them in aggregate as shown in this particular format.

5.2 Other Surveillance Tools

It is understood that the customer's SMF data is kept on-line for one day and then written out to tape(s) for longer-term storage. In addition to the standard exception reporting program outlined in this paper, it must be possible for the security officer to look at the detail records associated with a particular user, a particular terminal, a particular job, or a particular file, in order to produce in detail the time sequence of operations actually performed during the job or session. It is not suggested that detailed time

sequences of operation be performed for every user at all times; rather, it has been found necessary in order to in greater detail what is going on, to be able to examine the individual accounting records making up a job or a session, particularly for those job sessions which exhibit parameter values outside of the statistical bounds established by the surveillance program.

In the case of the SMF records, it is possible for a user to spawn batch jobs from the VM system. It must be possible for all of the activities of a given user to be traced to the various machines which may be used in accomplishing his or her work. The experience with the model system indicates that it is important that the records making up a session or a job or a unit of work be presented contiguously rather than intermixing the records on the basis of an arbitrary time stamp associated with each record. In practice, this may mean detail entries will be tracked on the VM system to the point where a job is batched to the JES3 job distribution system, then through all the job steps of the batched job, and then back to VM to show the continuation of the activities on the VM in parallel with or while the batch job was running one or more of the batch systems.

In general, there is a requirement to be able to track jobs or sessions based on a variety of kinds of information; for example, terminal identifiers or specific devices referred to and the like. The requirement is to be able to either show all records with the same terminal identifier or the same device address, or sometimes to use the terminal identifier device address or other characteristics to identify the job and then to show all details for that particular job.

For instance, if there is reason to suspect that there is unwarranted file access activity against a particular file, one may wish to examine all details of activity against that file regardless of the individual programs making the references, in which case the file id would act as a pointer into the first SMF record that contained its identifier. From that record, the job identifier would be obtained and then the detail for· the entire job could be displayed or acquired.

5. 3 Summary

The computer base security audit and surveillance system can be an effective tool in security control and management of ADP resources. User, data set, and program profiles can provide security personnel with information regarding exceptional use of the system. While it is expected that nearly all such exceptional use will be benign, this approach makes it possible to detect possible misuse of the system. It gives security personnel important automated tools to help provide early detection of unauthorized malicious activity directed against ADP assets.

In the preceding sections, an outline of a system design and the basis for providing statistical detection of abnormal use was developed. The surveillance and detection system is a filter screening out the mass of users of any system who are not doing anything untoward. In general, what constitutes "abnormality" is parametric. It can be set for any given environment. While the bulk of the report focused on the identification of abnormal use by individual users, statistics similar to those described for individual users can be accumulated for the user population as a whole, and the entire population screened for the purpose of identifying potential detailed.

With the use of statistical parameters such as those described above, the system can report abnormalities; that is, usage outside of the range of those parameters. This does not mean that a particular episode involves anything wrong it merely means that something is statistically different from previous accumulated use of the system for that entity; that is, user, file, program, and so forth. If abnormal symptoms do not recur, it is likely that nothing much is happening; however, if the symptoms continue to show up, then the subject involved could be investigated further by more conventional means.

In any real-life situation, computer systems often have thousands of users and tens of thousands of programs in data files. It is necessary to reduce the volume of history data implied by these numbers in various ways. First, if there are individuals whose use of the system is subject to surveillance because of the sensitivity of their jobs or for any other reason, he or she becomes a subject of interest. The selection of job (that is, session, tasks, runs, etc.) records can and should be made on that user's identity to include such individuals. The system designs sketched in the preceding sections indicate the use of such selection functions.

Note that most of the tests applied to systems use are equally applicable to specific files, and, as the section indicated, one could use a pre-pass to collect user's identification for those users referring to a specific named object: file, device, system, and the like. Rather than attempt to treat all members of a large population with this system, at all times, a sampling technique can be applied to select subsets of the total population for examination either over a particular period of time such as two weeks or for a gross examination against gross parameters established for the population as a whole. Of the two approaches, the detail examination for several weeks appears a priori to be the preferred method.

6. Development Plans

6.1.Introduction

This section outlines a development plan and gives an estimated schedule and-level of effort to provide an operationally useful security surveillance system. No serious attempt has been made to estimate computer time or storage cost as this will be affected by the actual system configuration used to implement the design.

The basic system consists of two programs:

- Security Surveillance Subsystem
- Security Trace Subsystem

6.2 Surveillance Subsystem Functional Description

The Surveillance Subsystem will consist of three preparation steps and a series of report formatters. The function of this subsystem is to provide exception reports of "abnormal" system use by specified individuals.

The function of the first step of the surveillance subsystem is to extract from the dump data set all relevant SMF records associated with a list of users making up the (a) "watch list". The selected SMF records are collected in a single data set where they are sorted in time-sequence order by user-id.

The sorted selected records will be processed by the next step to create one record per job or session. The record will be identified by the user-id, and the list of data sets or files referred to as a job/session characteristic.

Detailed measures of time, I/0 activity, and the like, associated with the job/session (as described in section 3), will be collected in summary form in the job/session record.

(NOTE: Some of this data was apparently being collected in customer developed SMF records type 210 in 1978 and 1979. If these records are still being collected, this step mav merely be an adaptation of the program that produces the type 210 records.)

The job/session records will then be posted in user-id, job/session characteristic order for the update step to follow.

- The update step matches job/session records against history records to:
- determine whether individual job/session records are within statistical "normality";
- accumulate additional data to refine the statistics;
- look for single "abnormal" events (illegal logons, single parameter absolute values exceeding arbitrary thresholds, etc.);
- create "new" history records (existing user, new job/session characteristic or totally new user);
- drop "old" history records for lack of activity.

The update step will produce an exception file with all major exceptions reported at least by type (e.g., values exceed absolute limits; values exceed statistical limit; new records added; old records dropped for lack of activity; etc.).

The final step(s) are a set of report formatters that select a particular exception type and edit and format a report for that kind of exception (see Figures 6, 7, 8, and 9 for examples).

6.3

Tasks

	Tasks	Level of Effort (man-weeks)	Elapsed Time (weeks)
I.	Design Job/Session Record, History Record, and Exception Records	4	4
II.	Design Selection Step Program	1	1
III.	Design Job/Session Summary Program	2	2
IV.	Design Update Program	2	2
V.	Design Report Programs	1 (for 4 reports)	1
VI.	Code and Test Selection Step	2	2
VII.	Code and Test Summary Step	4	4
VIII.	Code and Test Update Step	8	8
IX.	Code and Test Exception Reports (approximately 4)	2	2
	TOTALS	26	26

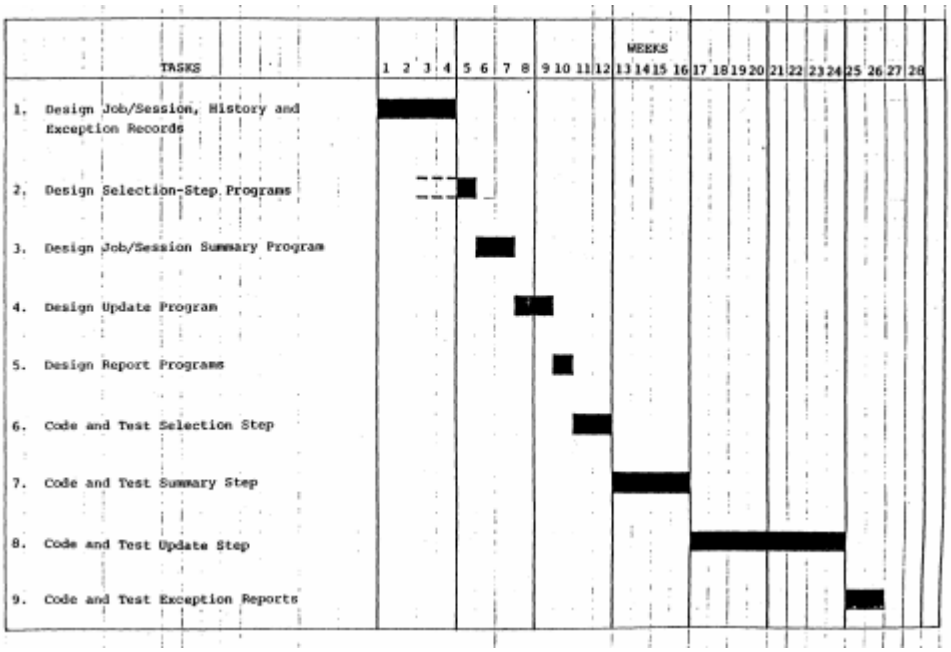


Figura 11: Summary Task Schedule for Security Surveillance Subsystems

6.4 Trace Subsystem Functional Description

The function of the trace subsystem is to produce from the SMF records a detailed, time-sequenced log of activity by (or on) a selected entity.

The Security-Trace Subsystem will accept parameters specifying the type of entity and the time scope of the trace. The trace report will be fixed for a given type of entity.

Parameters to the trace should include:

- Type of entity (job--id, user-id, data set, device-id, etc.);
- Time parameters:

  start date {if omitted - today)

  [end date] (if omitted - today)

  start time {if omitted - 00:00:00)

  [end time] {if omitted - 23:59:59)

As long as the times specified are increasing (and not overlapping), it should be feasible to trace multiple time ranges in a single pass of the "raw" SMF data.

Some time parameters might look like:

3/18/80
3/18/80 1600
3/18/80 - 3/20/80 1600
3/18/80 1600- 1830, 3/20/80 14:30 …

The trace records will have a standard part, then specific information that is appropriate to the record. A sample trace might look like:

TRACE FOR USER JONES.J
<DATE (OR DATE RANGE)>

TIME (HH:MM:SS.hh)	REC. TYPE		
15:23:01.00	JOB INIT	< JOB NAME >	
15:23:02.18	RACF PROC	JOB INIT	<job name>
15:23:07.46	RACF PROC	ACCESS	<data set name> <type of access> OLDF.DATA READ
15:23:17.49 ...			
.			
.			
.			
15:26:01.89	STEP TERM	< JOB NAME ><step name>...	
15:26:11.35	JOB TERM	< JOB NAME ><completion code>...	

6.5

Tasks

Tasks	Level of Effort (man-weeks)	Elapsed Time (weeks)
I. Design content of: . user-id trace . job-id trace . device-id trace . .	6	6
II. Design Trace Program	3	3
III. Code and Test Trace Program	3	3
TOTALS	12	12

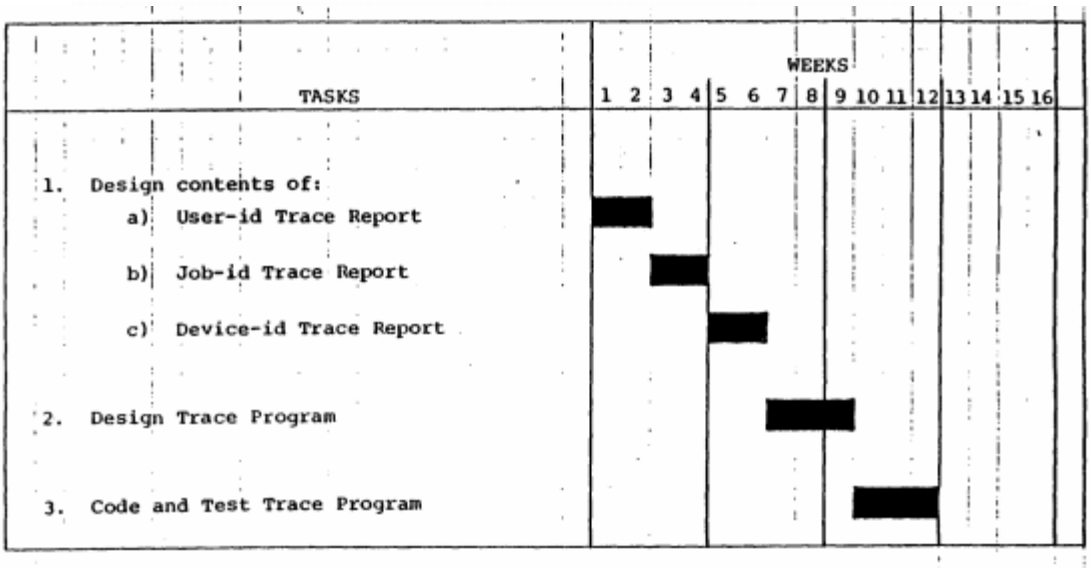


Figura 12: Summary Task Schedule for Security Trace Subsystem

6.6 Integration of Subsystems

The scope of this task depends on the system environment in which the security officer subsystems will be placed. If the programs are placed on the VM system, then one or more JCL sets (procedures) can be used to permit the programs to work with current SMF data (SYSl.MANX, SYSl.MANY data sets) or the dump data sets (SMF.DAILY.DATA) or the weekly data sets (SMF.WEEKLY.DATA). Allocation of the correct data sets can be done from the date parameters to the trace programs. There is no particular allocation required for the surveillance subsystem.

If the security officer surveillance subsystem(s) is placed on a standalone minisystem (for example), there is some action needed to either copy the entire dump data set to the minisystem (not recommended due to its size) or run the job/session select program on VM to produce a data set that will be brought over to the mini for processing.

Since access to current and recent SMF.DAILY.DATA and SMF.WEEKLY.DATA sets is needed for the trace function, and since at least the surveillance subsystem selection step must access the current SMF.DAILY.DATA, it appears that the security subsystem(s) should be placed in/on VM.

Tasks	Level of Effort (man-weeks)	Elapsed Time (weeks)
I. Define Integration Requirements	2	2
II. Code and Test Procs for Integration	2	2
TOTALS	4	4

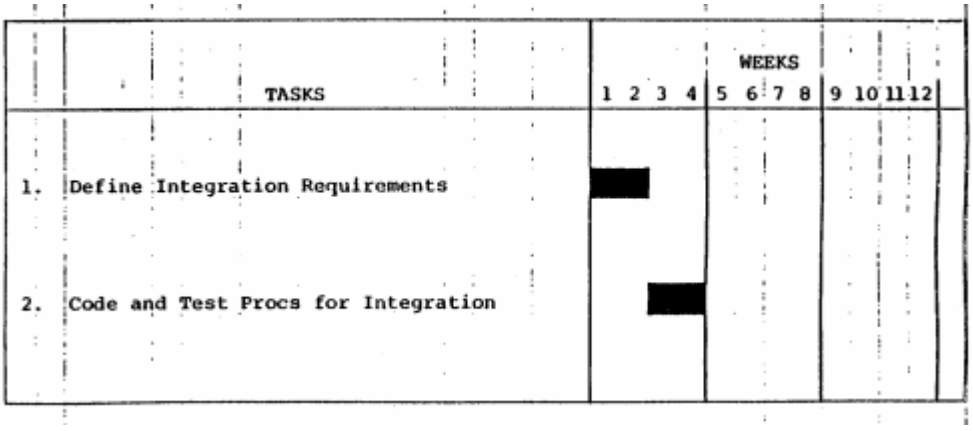


Figura 13: Summary Task Schedule for Integration Security Subsystem